

DESARROLLO DE UN SISTEMA DE GESTIÓN DE LA ELECCIÓN DE DOCENCIA

**DAVID ARROYO SEGOVIA
JAVIER PICATOSTE ZANGRÓNIZ**

**GRADO EN INGENIERÍA INFORMÁTICA
FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID**



**TRABAJO DE FIN DE GRADO
CURSO 2018-2019**

**DIRECTORES:
PABLO RABANAL BASALO
ANTONIO SARASA CABEZUELO**

A nuestros padres, hermanos y abuelos.

A Alba y Lucía.

AGRADECIMIENTOS

En primer lugar, nos gustaría dar las gracias a nuestros directores, Pablo Rabanal Basalo y Antonio Sarasa Cabezuelo, por habernos guiado durante el transcurso de este trabajo y por todos los consejos que nos han ido aportando. Probablemente, sin su ayuda este proyecto no habría sido posible.

Así mismo queremos agradecer a nuestras madres María José y Ana Blanca, y a nuestros padres Juan Carlos y Rafael todo el apoyo de valor incalculable que nos dais cada día. Sin vuestra comprensión y apoyo nada de esto se habría podido lograr. A nuestros hermanos Álvaro y Gonzalo, que se han interesado en todo momento por nosotros. A nuestros abuelos que siempre han querido lo mejor para nosotros.

También nos acordamos de nuestras novias Alba y Lucía, por soportarnos, hacernos reír y ayudarnos a mantener la cordura en los momentos más complicados.

RESUMEN

Este trabajo responde a la necesidad de automatizar todas las tareas que se realizan de manera manual para elegir la docencia de los profesores cada año en el departamento de Sistemas Informáticos y Computación de la Facultad de Informática de la Universidad Complutense de Madrid.

Actualmente, la elección de docencia se realiza de manera artesanal mediante hojas de cálculo. Debido a la complejidad de esta tarea, en este trabajo fin de grado se ha desarrollado un sistema que tiene como objetivo automatizar en cierto grado el proceso de elección de docencia. Para ello cada profesor tiene un usuario desde el cual puede elegir su docencia de acuerdo con las normas de elección del departamento. Por su parte, el departamento puede incluir la docencia a elegir de una manera sencilla mediante la lectura de hojas de cálculo. Así mismo, el sistema controla los problemas de incompatibilidad de horarios, y el resto de las restricciones que existen en el proceso de elección.

Para su implementación, se ha desarrollado una aplicación web. Además, se ha implementado un servicio REST que permita en el futuro la utilización del sistema en otro tipo de clientes.

Palabras clave

Planificación docente, aplicación web, gestión de docencia, reparto docente, API REST

ABSTRACT

This work responds to the need to automate all tasks that are performed manually to assign courses to teachers each year in the department of Computer Systems and Computing of the Faculty of Computer Science at the Complutense University of Madrid.

Currently, the assignment of courses is done by hand using spreadsheets. Due to the complexity of this task, a system has been developed in this end-of-grade work that aims to automate to a certain degree the process of choosing teaching. To do this, each teacher has a user from whom they can choose their teaching according to the rules of choice of the department. For its part, the department can include teaching to choose in a simple way by reading spreadsheets. Also, the system controls the problems of incompatibility of schedules, and the rest of the restrictions that exist in the election process.

For its implementation, a web application has been developed. In addition, a REST service has been implemented that will allow the system to be used in the future by other types of customers.

Keywords

Teacher planning, web application, teacher management, teacher distribution, API REST

ÍNDICE

RESUMEN	6
Palabras clave	6
ABSTRACT	7
Keywords	7
ÍNDICE DE FIGURAS	12
ÍNDICE DE TABLAS	15
1. INTRODUCCIÓN	16
1.1. Motivación	16
1.2. Objetivos	16
1.3. Estructura de la memoria	17
1. INTRODUCTION	18
1.1. Motivation	18
1.2. Objectives	18
1.3. Structure of the report	19
2. ESTADO DEL ARTE	20
2.1. Sistemas de recursos empresariales educativos	20
2.2. Software de planificación de horarios	20
3. TECNOLOGÍA EMPLEADA	22
3.1. Django	22
3.2. openpyxl	22
3.3. virtualenv	22
3.4. Angular	22
3.5. HTML5 y CSS3	22
3.6 NPM	23
3.7. Base de datos	23
3.8. Apache	23
3.9. Git	23
3.10. Visual Studio Code	24
4. CASOS DE USO	25
4.1 Módulo profesores	25
4.2. Módulo asignaturas	33
4.3. Módulo elección de docencia	38

5. MODELO DE DATOS	43
5.1 Modelo E-R	43
5.2. Implementación de la base de datos	44
5.2.1. Asignaturas	44
5.2.2. Profesores	47
5.2.3. Elección de docencia	48
5.2.4. Tablas administrativas	50
6. ARQUITECTURA DE LA APLICACIÓN	52
6.1. Patrones de diseño	53
6.1.1. Capa de presentación	53
6.1.2. Servicio REST	53
7. DISEÑO DE LA APLICACIÓN	54
7.1. Colores y tipografía	54
7.2. Funcionalidad de la aplicación	55
7.2.1. Departamentos	57
7.2.1.1. Dashboard	57
7.2.1.2. Configuración del departamento	58
7.2.2. Asignaturas	60
7.2.2.1. Lista de asignaturas	60
7.2.2.2. Detalle de la asignatura	63
7.2.2.3. Añadir o editar asignatura	64
7.2.2.4. Importar asignaturas	66
7.2.3. Profesores	67
7.2.3.1. Lista de profesores	67
7.2.3.2. Detalle del profesor	70
7.2.3.3. Añadir o editar profesores	73
7.2.3.4. Importar profesores	75
7.2.4. Elección de docencia	75
7.2.4.1. Listado de asignaturas y filtros	78
7.2.4.2. Información y mensajes	82
7.2.4.3. Calendario	82
7.2.4.4. Guardar o confirmar la docencia	85
7.2.5. Autenticación	87
7.2.5.1. Login y gestión del token	87
7.2.5.2. Cerrar sesión	89

7.2.5.3. Restricciones en la navegación	89
7.3 API REST	90
7.3.1. Asignaturas	92
7.3.2. Profesores	100
7.3.3. Elección de docencia	106
7.3.4. Departamentos	110
8. EVALUACIÓN	113
8.1. Detalles generales del encuestado	116
8.2. Cuestionario de satisfacción desde el punto de vista de un profesor	118
8.3. Cuestionario de satisfacción desde el punto de vista de un administrador	123
8.4. Nivel de satisfacción general	125
9. CONCLUSIONES Y TRABAJO FUTURO	126
9.1. Conclusiones	126
9.2. Trabajo futuro	126
9. CONCLUSIONS AND FUTURE WORK	128
9.1. Conclusions	128
9.2. Future work	128
10. APORTACIONES INDIVIDUALES	130
10.1. David Arroyo Segovia	130
10.2. Javier Picatoste Zangróniz	132
BIBLIOGRAFÍA	134
ANEXOS	137
Anexo I: Guía de uso	137
1. Iniciar sesión	137
2. Ver los detalles de una asignatura	138
3. Buscar asignatura	139
4. Ver los detalles de un profesor	140
5. Buscar profesor	141
6. Cambiar contraseña	142
7. Añadir asignatura a la elección	143
8. Enviar docencia al administrador	144
9. Buscar asignaturas desde la elección de docencia	145
10. Administrador - Añadir/Editar asignatura	146
11. Administrador - Añadir/Editar profesor	148
12. Administrador - Confirmar o eliminar docencia	150

13. Administrador - Cambiar configuración	151
14. Administrador - Iniciar y reiniciar docencia	152
15. Administrador - Importar y exportar lista de asignaturas	153
16. Administrador - Importar y exportar lista de profesores	154
Anexo II: Guía de instalación de la aplicación	157
1. Instalación de la aplicación web	157
2. Instalación de la API	158

ÍNDICE DE FIGURAS

Figura 1: Diagrama de casos de uso del módulo profesores	25
Figura 2: Diagrama de casos de uso del módulo asignaturas	34
Figura 3: Diagrama de casos de uso del módulo elección de docencia.....	38
Figura 4: Diagrama entidad-relación.....	43
Figura 5: Tablas de la base de datos de la aplicación Asignaturas.....	45
Figura 6: Tabla de la base de datos para la importación de asignaturas.....	47
Figura 7: Tablas de la base de datos de la aplicación Profesores	47
Figura 8: Tabla de la base de datos para la importación de profesores	48
Figura 9: Tablas de la base de datos de la aplicación Elección docencia.....	49
Figura 10: Tablas de la base de datos del núcleo de Django	50
Figura 11: Esquema representativo de la arquitectura del sistema.....	52
Figura 12: Fragmento de código - Códigos de color basados en el manual de estilo de la UCM.....	54
Figura 13: Fragmento de código - Fuentes de texto utilizadas en la aplicación web	55
Figura 14: Diagrama de módulos de la aplicación web	56
Figura 15: Pantalla del Dashboard.....	57
Figura 16: Fragmento de código - Gráfica de la pantalla de Dashboard HTML.....	57
Figura 17: Fragmento de código - Asignación de valores a la gráfica del Dashboard...	57
Figura 18: Página de configuración de la elección de docencia.....	59
Figura 19: Listado de las asignaturas	60
Figura 20: Panel de filtros de las asignaturas	61
Figura 21: Búsqueda de asignaturas por texto.....	62
Figura 22: Fragmento de código - Resaltar las asignaturas filtradas.....	62
Figura 23: Fragmento de código - Búsqueda de asignaturas.....	62
Figura 24: Detalle de la asignatura, primera pestaña.....	63
Figura 25: Detalle de la asignatura, segunda pestaña.....	63
Figura 26: Confirmación para borrar asignatura	64
Figura 27: Editar asignatura, paso uno	64
Figura 28: Editar asignatura, paso dos	65
Figura 29: Editar asignatura, paso tres	65
Figura 30: Editar asignatura, paso cuatro	65
Figura 31: Editar asignatura, paso cinco	65
Figura 32: Fragmento de código - Función para añadir/editar asignatura.....	66
Figura 33: Página para importar asignaturas	66
Figura 34: Fragmento de código - HTML de la página para importar asignaturas	67
Figura 35: Listado de profesores	68
Figura 36: Fragmento de código - Definición y vista de las columnas del listado de profesores	69
Figura 37: Filtros del listado de profesores	69
Figura 38: Búsqueda de profesores por texto	70
Figura 39: Detalle del profesor, primera pestaña	70
Figura 40: Detalle del profesor, segunda pestaña.....	71
Figura 41: Detalle profesor - Tarjeta asignatura normal	72
Figura 42: Detalle profesor - Tarjeta desdoble.....	72
Figura 43: Detalle profesor - Tarjeta asignatura divisible.....	72

Figura 44: Editar profesor, paso uno	73
Figura 45: Editar profesor, paso dos.....	74
Figura 46: Editar profesor, paso tres	74
Figura 47: Fragmento de código - Función para añadir o editar profesor.....	74
Figura 48: Fragmento de código - Generador de contraseñas seguras	75
Figura 49: Página de importar profesores.....	75
Figura 50: Fragmento de código - Petición para importar profesores.....	75
Figura 51: Página de la elección de docencia.....	76
Figura 52: Página elección de docencia - Estado de la elección minimizado	77
Figura 53: Página elección de docencia - Asignaturas minimizadas.....	77
Figura 54: Elección de docencia - Tarjeta de la asignatura normal.....	78
Figura 55: Elección de docencia - Tarjeta del desdoble	78
Figura 56: Elección de docencia - Tarjeta de la asignatura divisible	78
Figura 57: Fragmento de código - Elección de docencia - Función ejecutada tras seleccionar una asignatura normal.....	79
Figura 58: Fragmento de código - Elección de docencia - Actualizar elección	79
Figura 59: Fragmento de código - Elección de docencia - Comprobación de la docencia	80
Figura 60: Elección de docencia - Error en la elección.....	80
Figura 61: Elección de docencia - Filtros de las asignaturas.....	81
Figura 62: Fragmento de código - Elección de docencia - Asignatura disponible.....	81
Figura 63: Elección de docencia - Panel de información	82
Figura 64: Elección de docencia - Mensaje del profesor.....	82
Figura 65: Elección de docencia - Vista mensual del calendario	83
Figura 66: Elección de docencia - Vista semanal del calendario	83
Figura 67: Elección de docencia - Vista diaria del calendario	84
Figura 68: Fragmento de código - Elección de docencia - Configuración del calendario	84
Figura 69: Fragmento de código - Elección de docencia - Función de cambio del calendario.....	85
Figura 70: Elección de docencia - Conjunto de botones de un profesor	85
Figura 71: Elección de docencia - Mensaje para el administrador.....	86
Figura 72: Fragmento de código - Elección de docencia - Guardar o crear elección.....	86
Figura 73: Elección de docencia - Conjunto de botones de un administrador	86
Figura 74: Fragmento de código - Elección de docencia - Eliminar elección.....	87
Figura 75: Elección de docencia - Pantalla de login	88
Figura 76: Fragmento de código - Elección de docencia - Login	88
Figura 77: Fragmento de código - Elección de docencia - Añadir token a las peticiones	88
Figura 78: Fragmento de código - Elección de docencia - Fragmento del filtrado de errores	89
Figura 79: Fragmento de código - Elección de docencia - Logout	89
Figura 80: Fragmento de código - Elección de docencia - Ejemplo de restricción de las rutas	90
Figura 81: Fragmento de código - Elección de docencia - Rutas restringidas	90
Figura 82: Diagrama de clases de la API	91
Figura 83: Fragmento de código - Asignaturas serializer	96

Figura 84: Fragmento de código - Asignaturas viewset	97
Figura 85: Hoja de cálculo de las asignaturas	98
Figura 86: Fragmento de código - Importar asignaturas	99
Figura 87: Ejemplo de fixture para la importación de asignaturas	100
Figura 88: Correo electrónico confirmación registro con contraseña temporal	103
Figura 89: Fragmento de código - Enviar contraseña temporal	103
Figura 90: Fragmento de código - Cálculo de PDA y créditos fijos del departamento	105
Figura 91: Hoja de cálculo de los profesores	105
Figura 92: Fragmento de código - Actualización variables departamento	108
Figura 93: Fragmento de código - Comprobación solapamientos entre asignaturas....	109
Figura 94: Fragmento de código - Incompatibilidades con asignatura	109
Figura 95: Fragmento de código - Cálculo de variables del departamento y actualización de los créditos de los profesores	112
Figura 96: Formulario - Detalles generales del encuestado	113
Figura 97: Formulario - Pruebas desde el punto de vista de un profesor II	114
Figura 98: Formulario - Pruebas desde el punto de vista de un profesor III	114
Figura 99: Formulario - Pruebas desde el punto de vista de un profesor IV	114
Figura 100: Formulario - Pruebas desde el punto de vista de un profesor I.....	114
Figura 101: Formulario - Pruebas desde el punto de vista de un administrador	115
Figura 102: Formulario - Satisfacción diseño, fluidez y velocidad de la aplicación....	116
Figura 103: Evaluación - Edad de los encuestados	116
Figura 104: Evaluación - Sexo de los encuestados.....	117
Figura 105: Evaluación - Relación con la universidad del encuestado	117
Figura 106: Evaluación - Profesor - Pantalla de login.....	118
Figura 107: Evaluación - Profesor - Pantalla de dashboard	118
Figura 108: Evaluación - Asignatura - Menú	119
Figura 109: Evaluación - Profesor - Listado de asignaturas.....	119
Figura 110: Evaluación - Profesor - Filtrado de asignaturas	120
Figura 111: Evaluación - Profesor - Detalle de la asignatura.....	120
Figura 112: Evaluación - Profesor - Listado de profesores	121
Figura 113: Evaluación - Profesor - Filtrado de profesores	121
Figura 114: Evaluación - Profesor - Detalle del profesor.....	122
Figura 115: Evaluación - Profesor - Pantalla de elección de docencia	122
Figura 116: Evaluación - Profesor - Diseño del calendario de la elección de docencia	123
Figura 117: Evaluación - Administrador - Crear / Editar asignatura	123
Figura 118: Evaluación - Administrador - Crear / Editar profesor.....	124
Figura 119: Evaluación - Administrador - Elección de docencia.....	124
Figura 120: Evaluación – Diseño	125
Figura 121: Evaluación - Velocidad y fluidez.....	125
Figura 122: Pantalla de login.....	137
Figura 123: Pantalla de dashboard.....	138
Figura 124: Listado de asignaturas	138
Figura 125: Página de detalle de una asignatura	139
Figura 126: Vista del horario de la asignatura desde el detalle	139
Figura 127: Buscar asignatura	140
Figura 128: Listado de profesores	140

Figura 129: Página de detalle de un profesor	141
Figura 130: Vista de la docencia del profesor desde el detalle	141
Figura 131: Buscar profesor	142
Figura 132: Botón para cambiar la contraseña	142
Figura 133: Formulario para cambiar la contraseña	143
Figura 134: Añadir asignatura a la elección	143
Figura 135: Botones elección docencia	144
Figura 136: Mensaje para el administrador	144
Figura 137: Botón filtrar asignaturas desde la elección de docencia	145
Figura 138: Filtros de las asignaturas desde la elección de docencia.....	145
Figura 139: Botón de añadir asignatura.....	146
Figura 140: Botón editar asignatura	146
Figura 141: Editar asignatura	147
Figura 142: Añadir asignatura	147
Figura 143: Botón guardar edición o nueva asignatura.....	148
Figura 144: Botón para añadir profesor.....	148
Figura 145: Botón editar profesor	149
Figura 146: Editar profesor.....	149
Figura 147: Añadir profesor	150
Figura 148: Botón guardar edición o nuevo profesor.....	150
Figura 149: Confirmar la docencia.....	151
Figura 150: Cambiar configuración del departamento	151
Figura 151: Iniciar elección de docencia	152
Figura 152: Botón para reiniciar la elección de docencia	152
Figura 153: Popup para reiniciar la elección de docencia	153
Figura 154: Botón para exportar asignaturas.....	153
Figura 155: Botón para importar asignaturas	154
Figura 156: Página para importar asignaturas	154
Figura 157: Botón para exportar profesores	155
Figura 158: Botón para importar profesores.....	155
Figura 159: Página para importar profesores	156
Figura 160: Creación de la estructura de la base de datos.....	159
Figura 161: Creación de superusuario	160

ÍNDICE DE TABLAS

Tabla 1: Endpoints API - Asignaturas	92
Tabla 2: Parámetros de consulta API - Asignaturas	93
Tabla 3: Endpoints API - Profesores	100
Tabla 4: Parámetros de consulta API - Profesores	101
Tabla 5: Endpoints API - Elección de docencia	106
Tabla 6: Endpoints API - Departamentos.....	110

1. INTRODUCCIÓN

En este primer capítulo de la memoria se va a explicar la motivación de la aplicación desarrollada, cuáles han sido sus objetivos y se describirá la estructura de la memoria.

1.1. Motivación

La elección de docencia es un proceso complicado y que hasta ahora suele llevar alrededor de media semana en finalizarse. Se hace de manera artesanal, con hojas de cálculo en las que abundan las fórmulas e instrucciones complejas similares a las de un lenguaje de programación. Además, una de las razones de la lentitud del proceso es que, debido a la propia naturaleza de la hoja de cálculo, cuando un profesor elige docencia se lo tiene que comunicar al encargado de gestionar el proceso, éste debe comprobar que no hay incompatibilidades y después confirmar al profesor la validez de su elección.

En ocasiones, es complicado que el profesor conozca de antemano las incompatibilidades que existen en su elección, para comprobarlo debe estar en continuo contacto con el coordinador de la elección. Algunos años se hacían reuniones en las que participaban todos los miembros del departamento para repartirse la docencia, sin embargo, estas reuniones eran muy largas.

En este trabajo se ha desarrollado una herramienta que implementa un sistema de elección de docencia basado en el escalafón, donde el orden de elección de un profesor está condicionado por el tipo de puesto de trabajo que tiene y por la antigüedad en el puesto.

1.2. Objetivos

El objetivo principal de este proyecto era desarrollar un sistema que permitiera gestionar el proceso de elección de docencia del departamento de Sistemas Informáticos y Computación de la Facultad de Informática de la Universidad Complutense de Madrid. Los objetivos específicos planteados en el proyecto fueron los siguientes:

- Desarrollar una aplicación web que permita a los profesores elegir su docencia, ver todas las asignaturas que tiene a su disposición y comprobar las incompatibilidades y restricciones que existen en su elección. Así mismo, el encargado de coordinar la elección de docencia podrá observar en todo momento las estadísticas del proceso de elección y gestionar las diferentes incidencias que puedan surgir en el proceso interviniendo cuando sea necesario.
- Implementar una API RESTful que contenga toda la lógica de negocio y reciba las peticiones de la aplicación web.
- Establecer un sistema que permita extraer la información inicial del departamento de hojas de cálculo y sea capaz de importarla al sistema.
- Permitir al administrador del sistema la configuración de la aplicación a las necesidades particulares que puedan surgir en cada elección de docencia.

1.3. Estructura de la memoria

La memoria de este proyecto se estructura de la siguiente manera. En el capítulo 1 se introduce el trabajo realizado y la motivación para realizar su desarrollo. Así mismo se describe la estructura de la memoria.

En el capítulo 2 se revisan algunas herramientas con una funcionalidad semejante a la desarrollada en este trabajo.

En el capítulo 3 se comenta la tecnología empleada para la implementación.

El capítulo 4 se centra en los casos de uso de la aplicación, divididos en tres módulos: profesores, asignaturas y elección de docencia.

En el capítulo 5 se describe la forma en la que se ha realizado la persistencia de la información del sistema creado. En primer lugar, se expone el diagrama entidad-relación del sistema y la implementación del mismo en la base de datos. En cada apartado se adjuntan esquemas y figuras para facilitar la comprensión.

El capítulo 6 trata sobre la arquitectura y los patrones de diseño utilizados en la aplicación.

El diseño y funcionalidad de la aplicación se trata en el capítulo 7, dividido en dos secciones, en primer lugar, la aplicación web y, en segundo lugar, la API REST.

La evaluación por parte de usuarios encuestados, las conclusiones y el trabajo futuro, y las aportaciones individuales se desarrollan en los capítulos 8, 9 y 10, respectivamente.

En último lugar se encuentra la bibliografía y el anexo. Este último incluye una guía de uso, así como una guía de instalación tanto de la aplicación web como de la API.

1. INTRODUCTION

This first chapter of the report will explain the motivation of the application developed, what its objectives have been and will describe the structure of the report.

1.1. Motivation

The teaching assignment is a complicated process and so far usually takes about half a week to complete. It is done in a handmade way, with spreadsheets that abound with formulas and complex instructions similar to those of a programming language. In addition, one of the reasons for the slowness of the process is that, due to the very nature of the spreadsheet, when a teacher chooses the subjects, he or she must inform the person in charge of managing the process, who must check that there are no incompatibilities and then confirm to the teacher the validity of his or her choice.

Sometimes, it is difficult for the teacher to know beforehand the incompatibilities that exist in their choice, to verify this must be in continuous contact with the coordinator of the election. Some years there were meetings in which all the members of the department participated to share the teaching, however, these meetings were very long.

In this work, a tool has been developed that implements a teaching choice system based on the scale, where the order in which a teacher is chosen is conditioned by the type of job he or she has and by seniority in the position.

1.2. Objectives

The main objective of this project was to develop a system to manage the teaching selection process of the Department of Computer Systems and Computing of the Faculty of Computer Science of the Complutense University of Madrid. The specific objectives of the project were as follows:

- Develop a web application that allows teachers to choose their teaching, see all the subjects at their disposal and check the incompatibilities and restrictions that exist in their choice. Likewise, the person in charge of coordinating the choice of teaching will be able to observe at all times the statistics of the election process and manage the different incidents that may arise in the process, intervening when necessary.
- Implement a RESTful API that contains all business logic and receives requests from the web application.
- Establish a system that allows you to extract the initial information from the spreadsheet department and be able to import it into the system.
- Allow the system administrator to configure the application to the particular needs that may arise in each teaching choice.

1.3. Structure of the report

The report of this project is structured as follows. Chapter 1 introduces the work carried out and the motivation to carry out its development. The structure of the memory is also described.

Chapter 2 reviews some tools with similar functionality to the one developed in this paper.

Chapter 3 discusses the technology used for the implementation.

Chapter 4 focuses on the cases of use of the application, divided into three modules: teachers, subjects and teaching assignment.

Chapter 5 describes the way in which the persistence of the information in the system created has been carried out. First, the entity-relationship diagram of the system and its implementation in the database is presented. In each section, diagrams and figures are attached for ease of comprehension.

Chapter 6 deals with the architecture and design patterns used in the application.

The design and functionality of the application is dealt with in chapter 7, divided into two sections, firstly, the web application and secondly, the REST API.

Evaluation by surveyed users, conclusions and future work, and individual contributions are developed in chapters 8, 9 and 10, respectively.

In last place is the bibliography and the annex. The latter includes a user guide, as well as an installation guide for both the web application and the API.

2. ESTADO DEL ARTE

En el presente capítulo se recogerán las características más importantes acerca del software semejante que hay actualmente en el mercado.

La aplicación implementada es un sistema de planificación. En este ámbito se han encontrado dos tipos de software: los sistemas de planificación de recursos empresariales (ERP) especializados en el ámbito educativo y el software destinado a la planificación de horarios.

2.1. Sistemas de recursos empresariales educativos

Los ERP educativos están diseñados para la gestión integral de escuelas, colegios, institutos, centros privados y públicos, centros de formación, academias o cualquier otro tipo de centro dedicado a la formación. Cuentan con módulos para la gestión del personal, biblioteca, servicios estudiantiles, admisiones, donaciones, relación con egresados y finanzas. Sin embargo, estos sistemas suelen ser rígidos en cuanto a la configuración de sus servicios y no permiten su adaptación a contextos particulares. Algunos ejemplos de ERPs educativos son:

- Ellucian Banner Student [1]: Se trata de un ERP para gestionar centros educativos de diversa índole. Entre sus funcionalidades destaca:
 - Registro y monitoreo de estudiantes.
 - Herramientas de planificación académica.
 - Herramientas administrativas.
 - Definición de modelos académicos.
 - Calificaciones.
 - Generación de informes.
 - Planificación de horarios universitarios con restricciones personalizables.

- IFNOSS UCMS – University and College Management Software [2]: Es un sistema de planificación enfocado en la gestión de universidades de diverso tamaño. Permite interactuar entre los distintos órganos de las universidades estableciendo un canal de comunicación único. No tiene un sistema de planificación de horarios muy potente, y sus principales funcionalidades son:
 - Automatizar las admisiones permitiendo a los alumnos realizar la matrícula por Internet.
 - Realizar el pago de tasas.
 - Centralizar toda la información generada en un único sitio.
 - Calificaciones.
 - Gestión de recursos económicos y de personal.
 - Generación de métricas.
 - Planificación de horarios.
 - Control de asistencia y de tareas asignadas.

2.2. Software de planificación de horarios

Las aplicaciones de planificación de horarios se caracterizan porque son personalizables a nivel de restricciones, permiten la planificación de exámenes, la gestión

de aulas y son capaces de resolver las incompatibilidades que puedan aparecer. Algunos ejemplos de este tipo de aplicaciones son:

- UniTime [3]: Es un sistema de planificación de horarios optimizado para centros universitarios. Se puede utilizar para mantener la planificación horaria de las clases y/o de los exámenes. Fue originalmente desarrollada por docentes y estudiantes de varias universidades de Estados Unidos y Europa. Es destacable comentar que el proyecto está publicado bajo una licencia *open source* y animan a que otros estudiantes contribuyan con el proyecto. Sus características principales son:
 - Compatibilidad para gestionar horarios de distintas universidades y departamentos.
 - Planificación en períodos de exámenes.
 - Contempla diferentes categorías de profesores, créditos por asignatura y roles dentro de la aplicación.
 - Resolutor de restricciones.
 - Estadísticas de incompatibilidades.
 - Reportes.
- Mimosa Scheduling Software [4]: Es una aplicación que posibilita la planificación de horarios de manera automática. Entre sus funcionalidades se encuentran:
 - Planificación de horarios automáticamente teniendo en cuenta profesores, espacios disponibles, huecos entre horas, etc.
 - Generar sugerencias para optimizar los horarios.
 - Resolver conflictos.
 - Modificar manualmente el funcionamiento para tener en cuenta las restricciones particulares de cada organización.
 - Posibilidad de trabajar en la nube.

3. TECNOLOGÍA EMPLEADA

En este capítulo, se van a describir las herramientas tecnológicas empleadas en el desarrollo de este proyecto.

3.1. Django

Django es un *framework* de desarrollo web de código abierto [5], escrito en Python, que respeta el patrón de diseño conocido como Modelo-vista-templete. Su filosofía cuenta con principios como la reutilización, la seguridad, la extensibilidad de componentes, el desarrollo rápido y el principio No te repitas (DRY, del inglés *Don't Repeat Yourself*) [6]. Algunas de las ventajas que ofrece esta librería a la hora de implementar APIs son:

- API navegable desde el navegador facilitando el trabajo de los desarrolladores.
- Integración con autenticación basada en OAuth1 o OAuth2.
- Serialización de datos a partir de mapeo objeto-relaciones (más conocido por *Object-Relational mapping*, o sus siglas ORM).
- Extensa documentación y una comunidad activa al ser un proyecto *open source*.
- Empresas reconocidas como Red Hat o Mozilla lo utilizan.

3.2. openpyxl

La librería *openpyxl* [7] permite leer y modificar hojas de cálculo de Excel. Entre las funcionalidades más destacadas del modo escritura se encuentra la posibilidad de utilizar fórmulas, modificar filas y columnas, generar gráficas y definir formatos condicionales sobre las celdas. El modo lectura permite iterar sobre las filas o las columnas para extraer sus valores. Sin embargo la librería no permite leer todos los elementos que posibilita Excel, tales como imágenes o gráficas.

3.3. virtualenv

virtualenv [8] es una herramienta de desarrollo en Python que permite encapsular un proyecto para instalar paquetes sin interferir con otros entornos aislados ni con los paquetes de Python del sistema, evitando conflictos de paquetes en el intérprete principal.

3.4. Angular

Angular es un *framework* de código abierto para aplicaciones web desarrollado en TypeScript [9] mantenido por Google. Se utiliza para crear aplicaciones web de una sola página (*Single-page application*). La herramienta analiza el código HTML que contiene etiquetas personalizadas, obedece a las directivas de estos atributos, y une la entrada o salida de la página a un modelo representado por las variables estándar de JavaScript. Angular implementa el patrón modelo-vista vista-modelo (MVVM) [10] y ofrece un desarrollo orientado a componentes.

3.5. HTML5 y CSS3

Para la capa de presentación además se han utilizado HTML5, CSS3 en combinación con el metalenguaje Sass.

HTML5 [11] es la última versión del lenguaje de marcado para el desarrollo de páginas web. Esta versión establece un conjunto de nuevos elementos y atributos con significado semántico que manifiestan las nuevas necesidades de las páginas web modernas. HTML [12] describe el contenido de la página web, pero no su funcionalidad.

En este sentido usa marcado para estructurar el contenido, tales como imágenes o texto. Aunque puede describir, hasta un cierto punto, la apariencia de un documento, otras tecnologías son usadas generalmente para especificar la presentación de una página web.

CSS [13] es el lenguaje utilizado para describir la presentación de una página web. Una hoja de estilo CSS se compone de reglas que modifican la manera en que un documento, normalmente HTML, es presentado. Una regla CSS se compone de un selector, que seleccionará los elementos HTML que se actualizarán a través de una serie de propiedades.

Sass [14] es un metalenguaje de hoja de estilos que es traducido a CSS. Sass extiende CSS dotando al lenguaje de funcionalidades que están presentes en los lenguajes de programación orientados a objetos, tales como herencia de reglas, variables o funciones [15].

3.6 NPM

NPM [16] es un manejador de paquetes para el lenguaje de programación JavaScript y el entorno de ejecución Node.js. Consiste en una interfaz de línea de comandos, también llamada npm, y una base de datos online con paquetes públicos y privados, llamada *npm registry*. NPM administra las dependencias locales de un proyecto, como también las herramientas de JavaScript que estén instaladas en el sistema.

3.7. Base de datos

SQLite [17] es sistema de gestión de bases de datos relacional liviano, rápido y autocontenido. A diferencia de los sistemas de gestión de bases de datos tradicionales, que crean procesos independientes al del programa principal, el SQLite es una biblioteca que se enlaza con el software. Debido a esta característica, el programa se comunica con la base de datos a través de llamadas a funciones. Esto aminora la latencia de las comunicaciones, ya que las llamadas a funciones son más eficientes que la comunicación entre procesos [18].

3.8. Apache

El servidor HTTP Apache es un servidor web *open-source* multiplataforma [19]. Se caracteriza porque es un servidor estable, eficiente y extensible, capaz de trabajar con los últimos estándares HTTP y con la mayor parte de las extensiones web que existen en la actualidad (PHP, SSL, CGI, proxy, etc.). Además, dispone gran cantidad de módulos que extienden su funcionalidad.

3.9. Git

Como sistema de control de versiones se ha utilizado Git [20]. La utilización de un software de control de versiones permite a los desarrolladores trabajar en paralelo. Además, cuando los requisitos del proyecto cambian a menudo en ocasiones es necesario revertir a versiones anteriores del código. La característica de Git que realmente lo distingue de casi cualquier otro VCS (del inglés *Version Control System*) es su modelo de ramificación. Git permite tener múltiples ramas locales totalmente independientes entre sí que posteriormente pueden ser fusionadas.

3.10. Visual Studio Code

Como editor de código de fuente se ha utilizado Visual Studio Code [21]. Incluye soporte para la depuración, control de versiones y resaltado de sintaxis. Aunque no se trate de un entorno de desarrollo integrado, a través de distintas extensiones se puede lograr una experiencia cercana a un IDE [22].

4. CASOS DE USO

En este capítulo se van a especificar los casos de uso de la herramienta desarrollada en este trabajo de fin de grado. Los casos de uso se agrupan de acuerdo a los módulos que conforman la aplicación: profesores, asignaturas y elección de docencia.

Los actores definidos en los casos de uso que se muestran en este capítulo son: el profesor y el administrador del sistema. El primero puede visualizar la misma información que el administrador, salvo que solo tiene permisos de edición sobre su propia elección de docencia y perfil de usuario. En cambio, el administrador además de poder visualizar el mismo contenido que el profesor, tiene permisos de edición sobre cualquier contenido, incluyendo las elecciones docentes de los profesores. La figura del administrador será necesaria para inicializar el sistema e intervenir en el proceso de elección de docencia cuando sucedan eventos excepcionales que necesiten ser solventados.

4.1 Módulo profesores

El módulo profesores engloba toda la funcionalidad relativa a la gestión de los perfiles de los profesores y el sistema de autenticación. Como se ilustra en la figura 1, el profesor puede visualizar los perfiles o exportar el listado de todos los profesores, sin embargo, no tiene acceso a funciones que conlleven la edición o creación de nuevos perfiles.

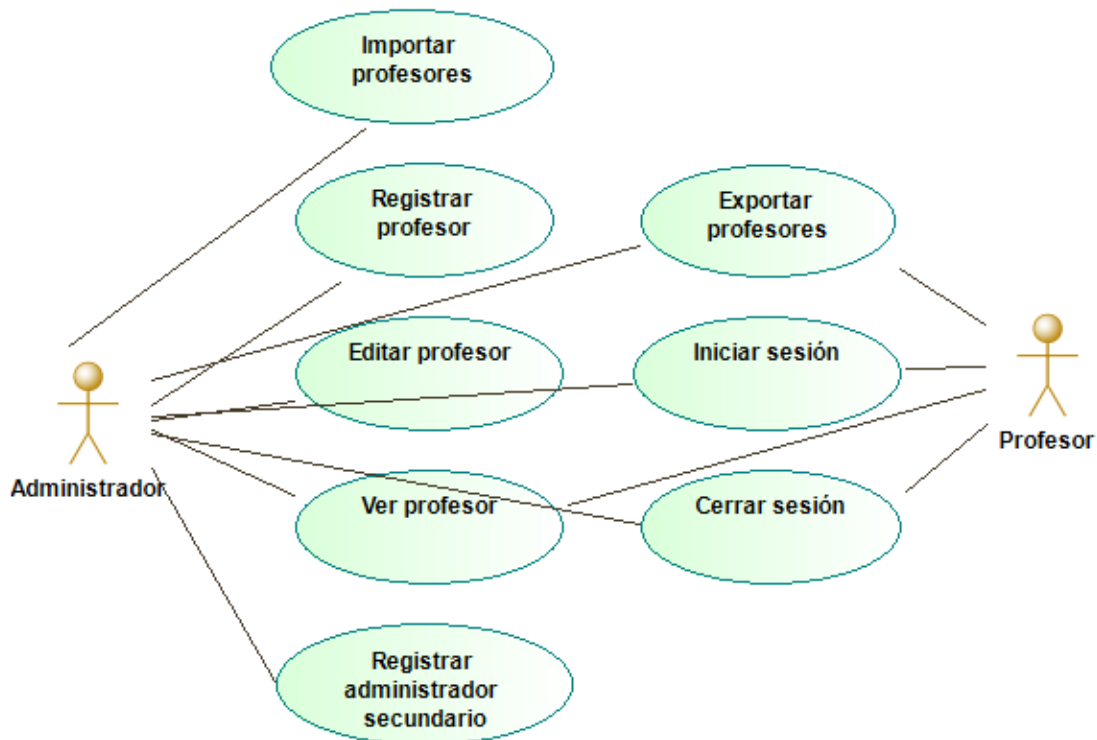


Figura 1: Diagrama de casos de uso del módulo profesores

A continuación, se listan los casos de uso asociados a este módulo.

DOCENCIA01	Importar profesores
Actor	Administrador
Descripción	Importa de un documento Excel el listado de profesores con su capacidad docente, PDAs, deudas, etc.
Precondición	Es necesario tener un documento Excel con el formato que impone la herramienta de planificación docente para que sea correctamente importado
Secuencia normal	<ol style="list-style-type: none"> 1. El sistema muestra un campo para subir el fichero 2. Se muestra una opción para sobrescribir la lista de profesores actuales o añadir nuevos al sistema 3. Subida del fichero .xlsx 4. El sistema informa por correo electrónico a los profesores de la contraseña temporal que se le ha asignado para poder acceder al sistema.
Postcondición	La lista de profesores se puebla con la información contenida en la hoja de cálculo
Excepciones	<p>Paso 2: Los profesores que se quieren importar ya se encuentran entre los profesores registrados</p> <p>Paso 3: La extensión del fichero no es la correcta o la maquetación de la hoja de cálculo no se ajusta a las normas de la herramienta</p>
Comentarios	

DOCENCIA02	Exportar profesores
Actor	Administrador y profesor
Descripción	Exporta a un documento Excel el listado de profesores
Precondición	Es necesario que se hayan añadido profesores con anterioridad al sistema
Secuencia normal	1. Se accede al listado de profesores y se presiona sobre el botón “Exportar profesores”
Postcondición	Se genera un fichero Excel .xlsx con el listado de profesores. El documento tendrá el mismo formato que debe tener cuando se importan
Excepciones	
Comentarios	

DOCENCIA03	Registrar profesor
Actor	Administrador
Descripción	Se registra un nuevo profesor en un departamento concreto
Precondición	Tiene que existir en el sistema el departamento al que pertenece el profesor
Secuencia normal	<ol style="list-style-type: none"> 1. El sistema muestra un formulario con algunos de los siguientes campos: <ol style="list-style-type: none"> a. Nombre del profesor b. Correo electrónico c. Departamento al que pertenece d. Tipo de contrato e. Escalafón f. Deudas de años anteriores g. PDA h. Permisos en el sistema i. Otra información personal relevante 2. El sistema informa de que el profesor ha sido registrado correctamente 3. El sistema envía un correo electrónico al profesor registrado informando de la contraseña temporal que se le ha asignado
Postcondición	El sistema registra un nuevo profesor el cual ya puede identificarse en el sistema y elegir docencia cuando esta sea iniciada
Excepciones	<p>Paso 1:</p> <p>Alguno de los campos introducidos no es correcto y habría que volver a pedir el usuario que vuelva a introducir los que no lo son</p>
Comentarios	Puede que también sea útil añadir la disponibilidad del profesor a la hora de registrarlo

DOCENCIA04	Editar profesor
Actor	Administrador
Descripción	Se editan algunos de los campos de un profesor
Precondición	El profesor que se vaya a editar tiene que haber sido registrado anteriormente y tener la cuenta habilitada
Secuencia normal	<ol style="list-style-type: none"> 1. El administrador selecciona de la lista de profesores mediante una búsqueda al profesor que quiere editar, se accede a la vista detallada y posteriormente a la ventana de edición 2. Se selecciona el campo que se quiere editar y se introduce el nuevo valor 3. El sistema avisa de la actualización del profesor
Postcondición	El profesor queda actualizado en el sistema con los nuevos valores introducidos
Excepciones	<p>Paso 1:</p> <p>Alguno de los campos introducidos no es correcto y habría que volver a pedir el usuario que vuelva a introducir los que no lo son</p>
Comentarios	Los campos que probablemente se actualicen con más frecuencia sean: las deudas, PDA y los relacionados con la posición en el escalafón de elección

DOCENCIA05	Ver profesor
Actor	Administrador y profesor
Descripción	El usuario accede a la información de un profesor para ver su información y su elección de docencia
Precondición	El profesor tiene que estar registrado en el sistema
Secuencia normal	<ol style="list-style-type: none"> 1. Se muestra una lista con todos los profesores del departamento 2. El administrador selecciona un profesor de la lista o hace una búsqueda del profesor a través de los filtros proporcionados 3. Se muestra la información del profesor con el estado actual de su elección de docencia
Postcondición	
Excepciones	<p>Paso 2:</p> <p>El valor buscado no coincide con ningún profesor registrado. El sistema informa de que compruebe el valor introducido</p>
Comentarios	

DOCENCIA06	Registrar administrador secundario
Actor	Administrador
Descripción	Registro de un nuevo administrador en el sistema que pueda tener algunas de las funciones del administrador principal
Precondición	El administrador principal tiene que estar identificado en el sistema
Secuencia normal	<ol style="list-style-type: none"> 1. Se accede al listado de profesores 2. Se accede a la vista detallada del profesor y se acciona sobre la opción de editar 3. Se acciona el interruptor que habilita al profesor para realizar tareas de administración 4. Se confirman los cambios
Postcondición	Hay un nuevo administrador en el sistema que tiene permisos para gestionar el sistema
Excepciones	
Comentarios	La motivación de crear un administrador secundario es la de que, en caso de ausencia del administrador principal, pueda haber una persona que pueda gestionar las incidencias y controlar el proceso de elección de docencia

DOCENCIA07	Iniciar sesión
Actor	Administrador y profesor
Descripción	El usuario accede a un formulario en el que introduce sus credenciales y se identifica en el sistema
Precondición	El usuario debe tener una cuenta anteriormente creada por un administrador y aún no estar identificado
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la vista de inicio de sesión o a una sección protegida 2. Introduce sus credenciales 3. El sistema le informa de que se ha identificado correctamente
Postcondición	El usuario queda debidamente identificado en el sistema pudiendo acceder a las vistas protegidas
Excepciones	<p>Paso 2:</p> <p>Las credenciales introducidas no son correctas o la cuenta que se intenta utilizar está deshabilitada</p>
Comentarios	Como trabajo futuro, sería deseable que la autenticación se realizará a través de la intranet del departamento

DOCENCIA08	Cerrar sesión
Actor	Administrador y profesor
Descripción	Se cierra la sesión en todas las ventanas que estén activas del navegador actual
Precondición	El usuario debe tener una cuenta identificada en el sistema
Secuencia normal	1. El usuario accede a la vista de cierre de sesión visible en todas las secciones de la aplicación
Postcondición	Se termina la sesión en el navegador actual
Excepciones	
Comentarios	

4.2. Módulo asignaturas

El módulo asignaturas comprende el conjunto de funcionalidades que afectan a la administración de las asignaturas disponibles durante el proceso de elección de docencia. Como se muestra en la figura 2, el administrador puede realizar tareas que impliquen la edición de asignaturas, mientras que el profesor solamente puede ver y exportar el listado de asignaturas.

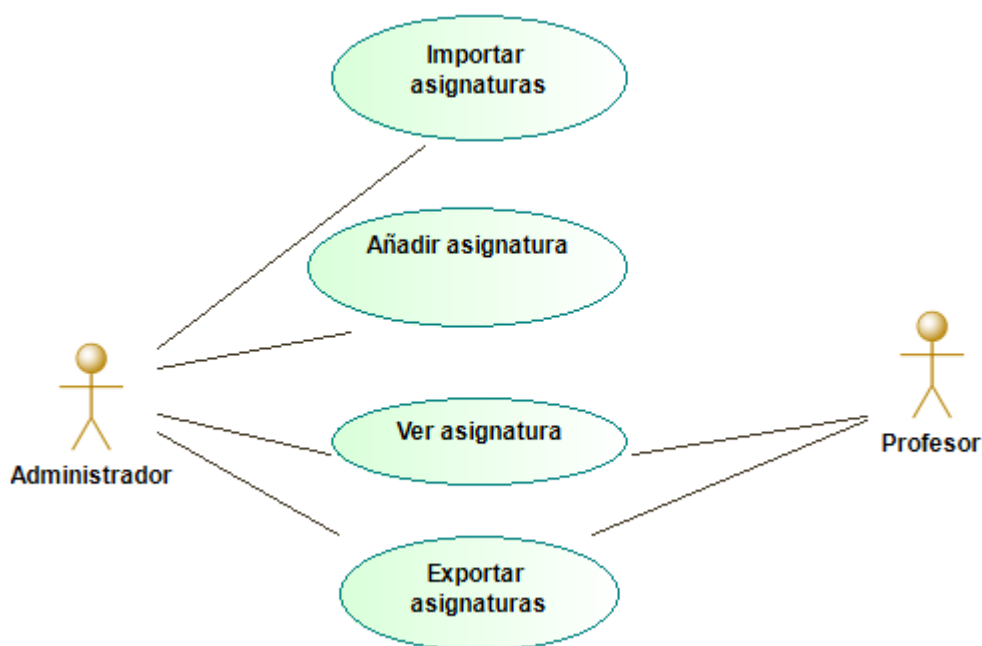


Figura 2: Diagrama de casos de uso del módulo asignaturas

A continuación, se listan los casos de uso asociados a este módulo.

DOCENCIA09	Importar asignaturas
Actor	Administrador
Descripción	Importa de un documento Excel el listado de asignaturas
Precondición	<p>Es necesario tener un documento Excel con el formato que impone la herramienta de planificación docente para que sea correctamente importado</p> <ul style="list-style-type: none"> ● Titulación ● Curso ● Código GEA ● Sigla ● Asignatura ● Grupo ● Departamento ● Cuatrimestre ● Horario ● Desdobles ● Aula y/o laboratorios
Secuencia normal	<ol style="list-style-type: none"> 1. Se rellenan los campos con las siglas del departamento al que pertenecen las asignaturas, el nombre completo del

	<p>departamento y el curso académico</p> <ol style="list-style-type: none"> 2. Se muestra una opción para sobrescribir la lista de asignaturas actuales o añadir nuevas al sistema 3. El sistema muestra un campo para subir el fichero 4. Subida del fichero .xlsx
Postcondición	El listado de asignaturas se puebla con la información contenida en la hoja de cálculo
Excepciones	<p>Paso 2: Las asignaturas que se quieren importar ya se encuentran entre las asignaturas registradas</p> <p>Paso 4: La extensión del fichero no es la correcta o la maquetación de la hoja de cálculo no se ajusta a las normas de la herramienta</p>
Comentarios	

DOCENCIA10	Exportar asignaturas
Actor	Administrador y profesor
Descripción	Exporta a un documento Excel el listado de asignaturas
Precondición	Es necesario que se hayan añadido asignaturas con anterioridad al sistema
Secuencia normal	<ol style="list-style-type: none"> 1. Se accede al listado de asignaturas y se presiona sobre el botón “Exportar asignaturas”
Postcondición	Se genera un fichero Excel .xlsx con el listado de asignaturas. El documento tendrá el mismo formato que debe tener cuando se importan
Excepciones	
Comentarios	

DOCENCIA11	Ver asignatura
Actor	Administrador y profesor
Descripción	A través de esta acción el profesor o administrador puede acceder a la información completa de la asignatura seleccionada
Precondición	La asignatura debe existir y el usuario debe estar identificado
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario ve la lista de asignaturas tanto en el momento de elección de docencia como en la página de las asignaturas 2. Selecciona la asignatura 3. Se le muestra toda la información relativa a la asignatura
Postcondición	
Excepciones	
Comentarios	Tanto en la vista con el listado de asignaturas como en la del proceso de elección de docencia existen filtros que facilitan la tarea de buscar una asignatura

DOCENCIA12	Editar asignatura
Actor	Administrador
Descripción	Edita los campos de una asignatura y permite añadir nuevos horarios y desdobles
Precondición	La asignatura debe existir y el usuario <i>logueado</i> debe tener permisos de administración
Secuencia normal	<ol style="list-style-type: none"> 1. Se accede a la vista detallada de una asignatura y posteriormente a la ventana de edición 2. Se modifican los campos, horarios y desdobles que se desee 3. El sistema informa de que se ha modificado la asignatura
Postcondición	La asignatura queda modificada en el sistema según los valores introducidos por el administrado
Excepciones	<p>Paso 2:</p> <p>Algunos de los campos modificados tienen valores incorrectos. El sistema informa de que se debe corregir el valor introducido</p>
Comentarios	

4.3. Módulo elección de docencia

El módulo de elección de docencia abarca las funcionalidades que permiten a los profesores decidir qué asignaturas incluyen en sus respectivas elecciones docentes. El administrador, como se ilustra en la figura 3, puede intervenir en el proceso y asignar la docencia de los profesores manualmente, así como configurar los parámetros del departamento que arbitran el proceso. Tanto el profesor como el administrador pueden visualizar las elecciones docentes de los demás profesores.

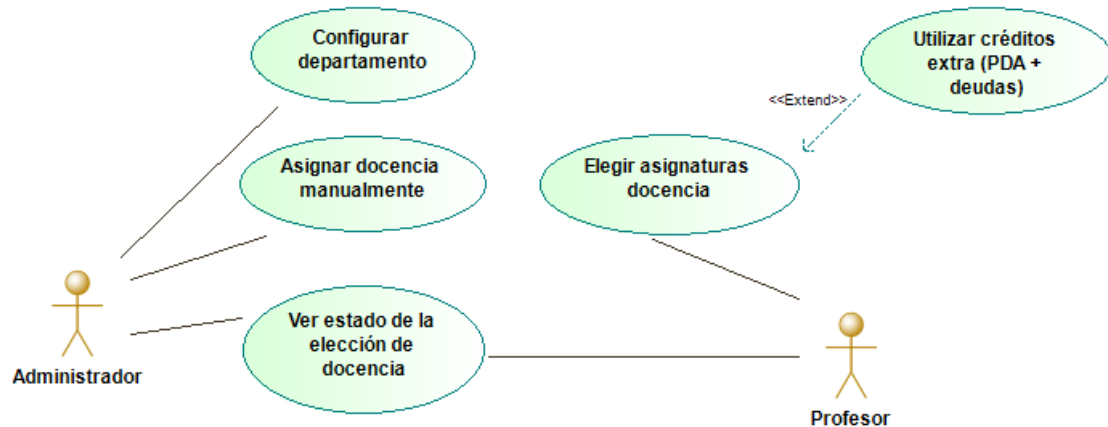


Figura 3: Diagrama de casos de uso del módulo elección de docencia

A continuación, se detallan los casos de uso que comprenden este módulo.

DOCENCIA13	Configurar departamento
Actor	Administrador
Descripción	Se especifican valores relativos al proceso de elección de docencia manualmente
Precondición	El administrador que haya iniciado sesión debe tener permisos de edición
Secuencia normal	<ol style="list-style-type: none"> 1. El sistema muestra un formulario con algunos de los siguientes campos: <ol style="list-style-type: none"> a. Nombre del departamento b. Total de créditos c. Capacidad docente del departamento d. Capacidad restante e. Carga dinámica 2. Se muestra una opción para iniciar el proceso de elección de docencia en el departamento 3. Se da la posibilidad de reiniciar el proceso de elección de docencia 4. El sistema informa de que el departamento se ha configurado correctamente

Postcondición	El sistema ha modificado los valores del departamento
Excepciones	Paso 1: Alguno de los campos introducidos no es correcto y habría que volver a pedir el usuario que vuelva a introducir los que no lo son
Comentarios	Es posible que al modificar manualmente los valores que condicionan el proceso de elección de docencia cause inconsistencias en el sistema

DOCENCIA14	Ver estado de la elección de docencia
Actor	Administrador y profesor
Descripción	El usuario accede a un resumen del estado actual de la elección de docencia. Se muestra información relativa a los créditos elegidos, profesores pendientes de elegir docencia, deudas y PDA totales, etc.
Precondición	El usuario debe estar <i>logueado</i>
Secuencia normal	1. El usuario accede a la vista resumen de la aplicación
Postcondición	
Excepciones	Paso 1: En caso de que el profesor aún no haya elegido asignaturas aparecerán los datos vacíos o un mensaje de aviso
Comentarios	Los datos se generarán de manera automática según avance el proceso de elección

DOCENCIA15	Asignar docencia manualmente
Actor	Administrador
Descripción	El administrador accede a una página prácticamente idéntica a la de elección de docencia individual de cada profesor, sin embargo, podrá editar toda la docencia ya elegida y asignar a cada profesor lo que desee
Precondición	El periodo de elección no debe haber empezado o debe haber terminado si no se trata de una incidencia
Secuencia normal	<ol style="list-style-type: none"> 1. El administrador ve la toda la docencia asignada y pendiente de asignar 2. Escoge el profesor al que desea modificar/asignar la docencia 3. Se seleccionan las asignaturas que se quieren añadir o eliminar de su docencia 4. El sistema analiza cada elección en busca de incompatibilidades
Postcondición	Se almacena la información con los cambios que el administrador ha realizado
Excepciones	<p>Paso 4:</p> <p>En caso de que el análisis de la elección encuentre algún conflicto se le notificará lo antes posible.</p>
Comentarios	El periodo debe haber terminado o no haber empezado para no encontrar problemas de concurrencia. Sin embargo, se puede confirmar la docencia con incompatibilidades adjuntando un mensaje explicativo

DOCENCIA16	Elegir asignaturas docencia
Actor	Profesor
Descripción	El profesor elige las asignaturas que impartirá durante el curso en la lista de asignaturas disponibles
Precondición	Todos los profesores de rango superior deben haber finalizado la elección de sus asignaturas
Secuencia normal	<ol style="list-style-type: none"> 1. El sistema muestra al usuario el listado de asignaturas disponibles 2. El usuario elige las asignaturas que quiere impartir 3. El sistema tras cada elección comprueba las incompatibilidades entre las asignaturas 4. El sistema informa de que la elección de docencia se ha realizado correctamente y se pasa el turno de elección al siguiente profesor
Postcondición	Quedará registrada la elección del usuario. Se le notifica al siguiente profesor que debe elegir asignaturas.
Excepciones	<p>Paso 3:</p> <p>En caso de que haya conflictos entre las asignaturas con respecto a las restricciones establecidas</p>
Comentarios	Se le informará al profesor si existen conflictos, si decide obviar el aviso se le notificará al administrador con el mensaje explicativo que ha adjuntado el profesor

5. MODELO DE DATOS

En este capítulo se va a describir la forma en la que se ha realizado la persistencia de la información del sistema creado.

5.1 Modelo E-R

En la figura 4 se muestra el diagrama entidad-relación que se ha utilizado para comenzar con el modelado de datos de la herramienta que se propone. En este diagrama se representan las entidades y los atributos definidos para el sistema: *Asignatura*, *Calendario*, *Horario*, *Departamento*, *Desdoble*, *AsignaturaDivisible*, *Profesor*, *EleccionDocencia*, *Deuda* y *Categoría*, así como las relaciones existentes entre ellas.

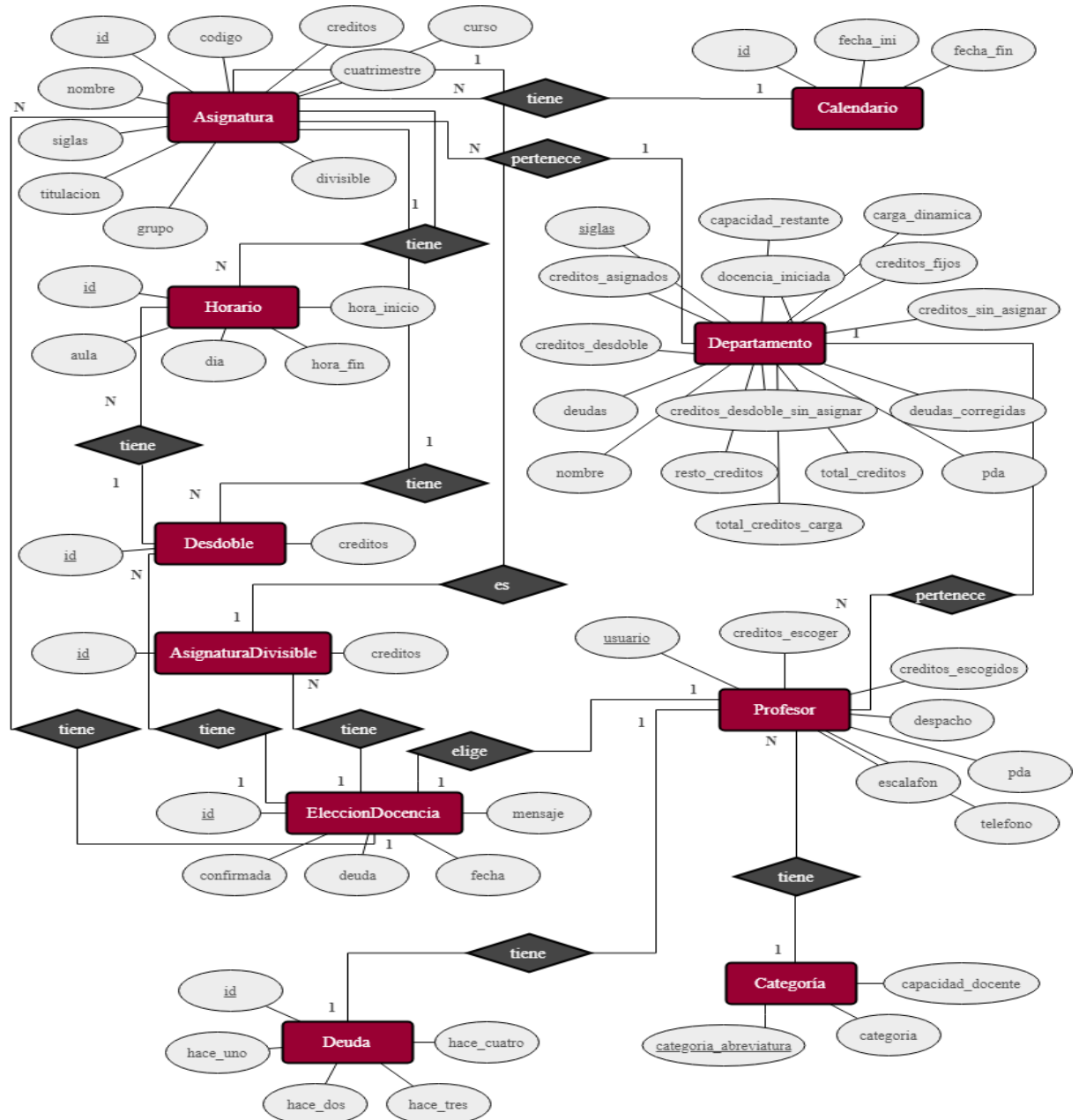


Figura 4: Diagrama entidad-relación

5.2. Implementación de la base de datos

Para implementar el modelo E-R descrito anteriormente se ha utilizado el modelo de datos de Django. En este *framework* una clase representa una tabla en la base de datos y las instancias de las clases son tuplas en la tabla. En este sentido Django proporciona una abstracción de la base de datos a través de una API que permite realizar las operaciones CRUD, aunque también es posible ejecutar sentencias SQL directamente. A continuación, se muestran las tablas utilizadas agrupadas por la información del tipo de entidad que almacenan.

5.2.1. Asignaturas

En este apartado se describen las tablas referidas a la información sobre las asignaturas disponibles en la aplicación. En la figura 5 se muestran las tablas que representan la información de las asignaturas registradas en la aplicación que pueden ser elegidas durante el proceso de elección de docencia. A continuación se describe cada tabla.

La tabla *asignaturas_asignatura* contiene la información de cada asignatura:

- *siglas*: Siglas de la asignatura.
- *nombre*: Nombre completo de la asignatura.
- *grupo*: Grupo al que pertenece.
- *cuatrimestre*: Cuatrimestre en el que se imparte.
- *codigo*: Código GEA de la asignatura.
- *curso*: Curso académico en el que se imparte.
- *titulacion*: Titulación a la que pertenece.
- *calendario_id*: Clave foránea del calendario asociado a la asignatura.
- *departamento_id*: Clave foránea del departamento al que pertenece.
- *creditos*: Créditos ECTS.
- *divisible*: Booleano que indica si se trata de una asignatura con posibilidad de tener docencia divisible.

La tabla *asignaturas_desdoble* contiene los desdobles asociados a una asignatura.

- *asignatura_id*: Clave foránea de la asignatura a la que pertenece el desdoble.
- *creditos*: Créditos ECTS asociados al desdoble.

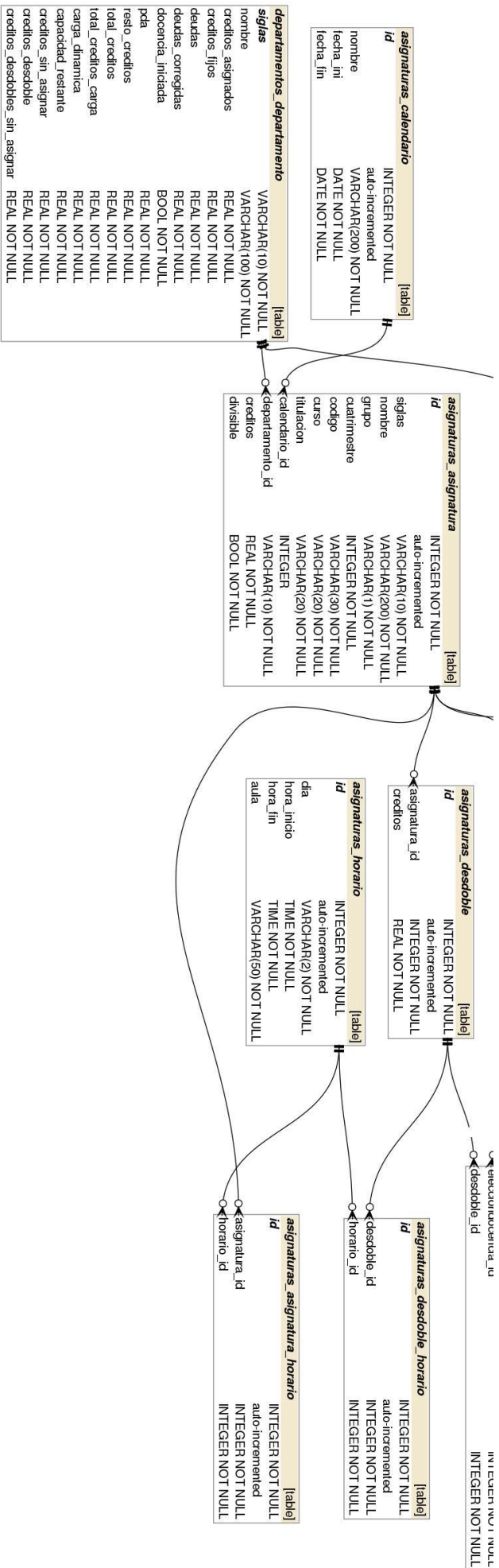


Figura 5: Tablas de la base de datos de la aplicación Asignaturas

La tabla *asignaturas_horario* representa el horario de una asignatura o de un desdoble (observar que las tabla *asignaturas_asignatura_horario* y *asignaturas_desdoble_horario* relacionan las asignaturas y los desdobles con sus respectivos horarios):

- *dia*: Día de la semana.
- *hora_inicio*: Hora de inicio de la clase.
- *hora_fin*: Hora de finalización de la clase.
- *aula*: Aula en la que se imparte la asignatura.

La tabla *departamentos_departamento* contiene los departamentos que gestiona la aplicación (el sistema está ideado para trabajar con un solo departamento pero sería fácilmente extensible a gestionar varios)

- *siglas*: Siglas del departamento.
- *nombre*: Nombre.
- *creditos_asignados*: Créditos de docencia asignada.
- *creditos_fijos*: Suma de créditos que se asignan antes del proceso de elección al personal ayudante.
- *deudas*: Suma de las deudas del personal docente del departamento.
- *deudas_corregidas*: Valor corregido según las normas del departamento de la suma de las deudas.
- *docencia_iniciada*: Booleano que indica si se ha iniciado el proceso de elección de docencia.
- *pda*: Suma de la PDA de todos los profesores.
- *resto_creditos*: Suma de todos los créditos del personal fijo.
- *total_creditos*: Total de créditos que debe asumir el departamento.
- *total_creditos_carga*: Suma de todas las capacidades docentes de los profesores.
- *carga_dinamica*: Carga dinámica del departamento teniendo en cuenta los créditos ya asignados y los que quedan por asignar del total.
- *capacidad_restante*: Capacidad restante del departamento.
- *creditos_sin_asignar*: Créditos que quedan pendientes de asignar:
- *creditos_desdoble*: Créditos totales de desdoble:
- *creditos_desdoble_sin_asignar*: Créditos de desdoble pendientes de asignar.

La tabla *asignaturas_calendario* contiene el calendario asociado a una asignatura. Es necesario para mantener una distinción entre las asignaturas anuales y cuatrimestrales. Además hay que tener en cuenta que no todas las titulaciones se rigen por la misma planificación docente.

- *nombre*: Nombre descriptivo del calendario.
- *fecha_ini*: Fecha de inicio.
- *fecha_fin*: Fecha de finalización.

<i>asignaturas_asignaturasimporter</i> [table]	
id	INTEGER NOT NULL auto-incremented
excel_file	VARCHAR(100) NOT NULL
departamento_siglas	VARCHAR(10) NOT NULL
departamento_nombre	VARCHAR(100) NOT NULL
sobrescribir	BOOL NOT NULL

Figura 6: Tabla de la base de datos para la importación de asignaturas

Para mantener un registro de las hojas de cálculo con las asignaturas que se han importado en el sistema se dispone de la tabla *asignaturas_asignaturasimporter* (ver figura 6).

- *excel_file*: Ruta del fichero Excel que se importa.
- *departamento_siglas*: Siglas del departamento al que pertenecen las asignaturas.
- *departamento_nombre*: Nombre del departamento.
- *sobrescribir*: Booleano para indicar si se han sobrescrito las asignaturas del departamento o se han incluido nuevas.

5.2.2. Profesores

A continuación, se detallan las tablas vinculadas con la información de los perfiles de los profesores. En la figura 7 se exponen las distintas relaciones que existen entre las tablas y los campos que contienen.

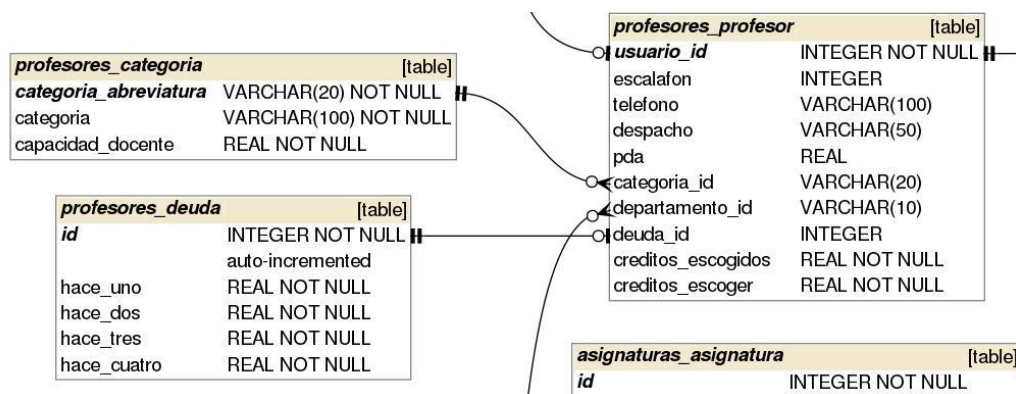


Figura 7: Tablas de la base de datos de la aplicación Profesores

En la tabla *profesores_profesor* están recogidos los diferentes profesores que participan en el proceso de elección de docencia. El campo *créditos_escoger* se calcula automáticamente en base a la categoría docente que tenga el profesor y el valor de carga dinámica que exista en el departamento durante la elección.

- *escalafon*: Posición del profesor en el escalafón del departamento. Las primeras posiciones eligen antes en el proceso de elección.
- *telefono*: Teléfono del profesor.
- *despacho*: Despacho del profesor.
- *pda*: Valor de PDA que posee el profesor que se suma a su docencia.
- *categoria_id*: Clave foránea de la categoría a la que pertenece.
- *departamento_id*: Clave foránea del departamento al que pertenece.

- *deuda_id*: Clave foránea de la deuda del profesor que puede descontarse de la docencia.
- *creditos_escogidos*: Créditos totales que ha seleccionado el profesor en su docencia.
- *creditos_escoger*: Créditos que debe escoger el profesor en su docencia en base a su categoría y la carga dinámica del departamento.

La tabla *profesores_categoria* almacena las diferentes categorías que pueden tener los profesores de un departamento.

- *categoria*: Nombre de la categoría (catedrático, contratado doctor, ayudante, etc.)
- *capacidad_docente*: Capacidad docente en número de créditos asociados a la categoría.

La tabla *profesores_deuda* recoge un listado de todas las deudas de los últimos cuatro años de los profesores. Esta deuda puede ser opcionalmente aplicada a su elección de docencia.

- *hace_uno*: Deuda del año anterior a la elección.
- *hace_dos*: Deuda de hace dos años.
- *hace_tres*: Deuda de hace tres años.
- *hace_cuatro*: Deuda de hace cuatro años.

<i>profesores_profesoresimporter</i> [table]	
<i>id</i>	INTEGER NOT NULL auto-incremented
<i>excel_file</i>	VARCHAR(100) NOT NULL
<i>departamento_siglas</i>	VARCHAR(10) NOT NULL
<i>sobrescribir</i>	BOOL NOT NULL

Figura 8: Tabla de la base de datos para la importación de profesores

Por último, la tabla *profesores_profesoresimporter* almacena las hojas de cálculo subidas al sistema para importar profesores (ver figura 8). En este sentido, mantiene un histórico de los ficheros subidos con la ruta del documento que se importó.

- *excel_file*: Nombre del fichero que se subió al sistema para importar los profesores contenidos en el mismo.
- *departamento_siglas*: Siglas del departamento al que pertenecen los profesores.
- *sobrescribir*: Booleano que indica si se sustituyeron los profesores por los del fichero o si se incorporaron nuevos a la lista.

5.2.3. Elección de docencia

En este apartado, se muestran las tablas que almacenan la información referente a las elecciones docentes de los profesores. En la figura 9 se pueden observar algunas de las relaciones que existen entre las distintas tablas, sin embargo, también existen relaciones con las tablas referidas a información sobre asignaturas y profesores.

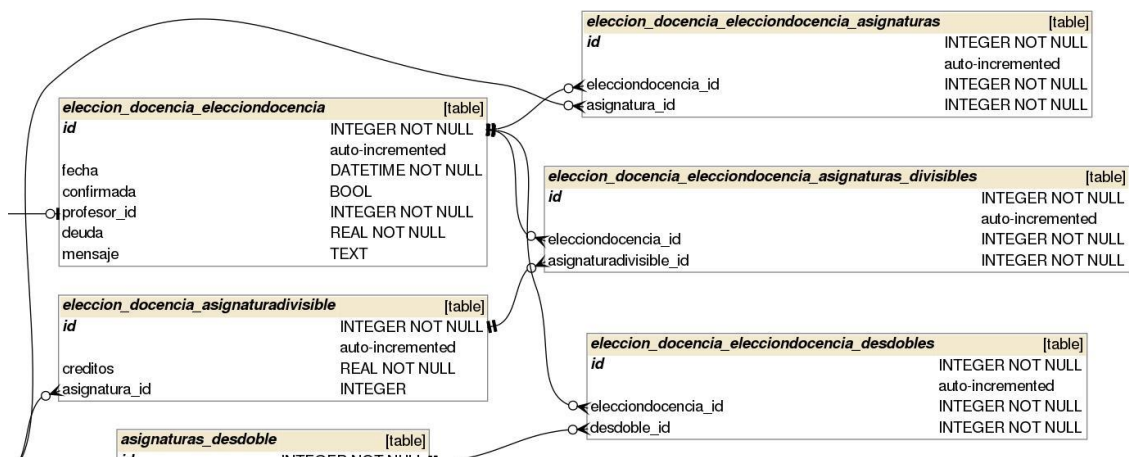


Figura 9: Tablas de la base de datos de la aplicación Elección docencia

La tabla *eleccion_docencia_elecciondocencia* recoge un listado de todas las elecciones de docencia de los profesores.

- *fecha*: Fecha en la que el profesor empezó a elegir docencia.
- *confirmada*: Indica si la docencia ha sido validada por un administrador del sistema.
- *profesor_id*: Clave foránea del profesor al que le pertenece la docencia.
- *deuda*: Cantidad de deuda del profesor que se ha decidido aplicar a la elección de docencia.
- *mensaje*: Mensaje opcional que se adjunta a la elección y que posteriormente el administrador podrá ver. El motivo de que el profesor pueda incluir un mensaje en su elección es el de poder argumentar decisiones que para el sistema son incompatibles o no válidas debido a casos límite que debe revisar el administrador.

La tabla *eleccion_docencia_elecciondocencia_asignaturas* contiene un listado de las asignaturas que se han escogido en las distintas elecciones de docencia del sistema.

- *elecciondocencia_id*: Clave foránea que referencia a la tabla que contiene las elecciones de docencia.
- *asignatura_id*: Asignatura escogida en la elección de docencia.

La tabla *eleccion_docencia_asignaturadivisible* recoge un listado de las asignaturas que tienen la posibilidad de ser impartidas por varios docentes y se han añadido a la elección como divisibles.

- *creditos*: Créditos que se escogen del total de créditos de la asignatura y que se añaden a la elección.
- *asignatura_id*: Clave foránea de la asignatura.

La tabla *eleccion_docencia_elecciondocencia_asignaturas_divisibles* almacena la relación entre asignaturas divisibles y docencias que las contienen. Únicamente tiene dos campos para mantener la relación.

Por último, la tabla *eleccion_docencia_elecciondocencia_desdobles* también registra las relaciones entre desdobles y docencias que incluyen desdobles en su elección.

5.2.4. Tablas administrativas

Las tablas que se listan en este apartado son las referentes al núcleo de Django, automáticamente generadas en el momento de la instalación del *framework*. La mayoría de las tablas almacenan información relativa a la gestión de usuarios y permisos. En la figura 10 se muestran las relaciones que hay entre las distintas tablas.

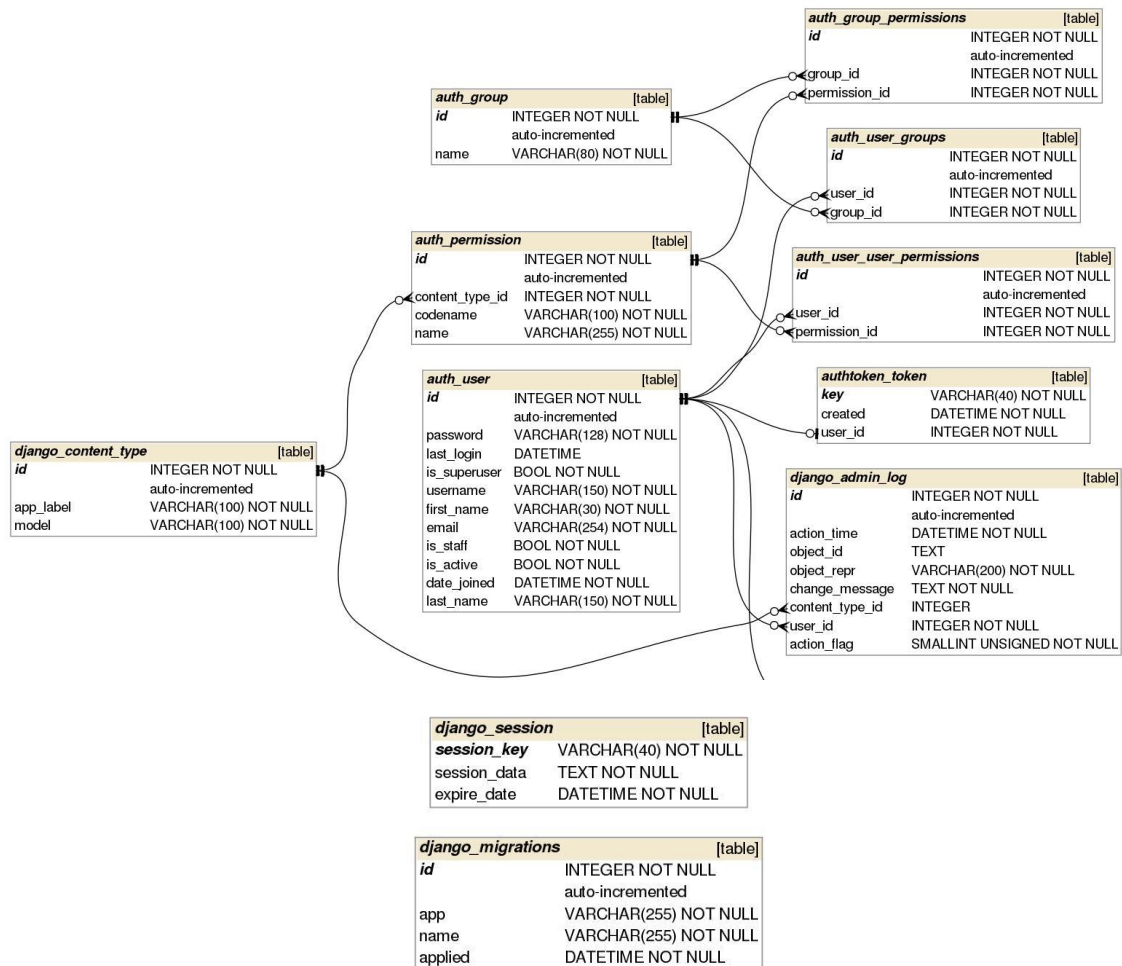


Figura 10: Tablas de la base de datos del núcleo de Django

La tabla *auth_user* es la más significativa de las del núcleo de Django dado que contiene los usuarios registrados en el sistema.

- *password*: Contraseña hasheada del usuario. Se utiliza el algoritmo PBKDF2 con un hash SHA256, método recomendado por el NIST [23].
- *last_login*: Fecha de la última identificación del usuario en el sistema.
- *is_superuser*: Booleano que indica si se trata de un superusuario.
- *username*: Nombre de usuario utilizado para iniciar sesión.
- *first_name*: Nombre del usuario
- *email*: Correo electrónico.
- *is_staff*: Booleano que indica si el usuario tiene permisos de administración.
- *is_active*: Booleano que indica si el usuario sigue activo en el sistema y puede identificarse.

- *date_joined*: Fecha en la que se registró el usuario en el sistema.
- *last_name*: Apellido del usuario.

6. ARQUITECTURA DE LA APLICACIÓN

En este capítulo se describe la arquitectura de la aplicación. En este sistema se ha utilizado un modelo cliente-servidor.

El modelo cliente-servidor es un modelo de diseño de software en el que los servidores ofrecen servicios a una serie de clientes que los consumen [24]. Los usuarios invocan la parte cliente de la aplicación, que realiza peticiones de servicio al servidor. El servidor recibe solicitudes, realiza los servicios requeridos y devuelve los resultados en forma de una respuesta.

En general, un servicio es una abstracción de una serie de recursos. El cliente para utilizarlos solo debe conocer el formato y contenido de la respuesta del servicio solicitado [25].

En la arquitectura de la aplicación se ha decidido implementar en el servidor una interfaz de programación de aplicaciones (API). La API es una capa de abstracción para acceder a un servicio. Lo que se consigue restringiendo a un formato específico la comunicación cliente-servidor es una mayor facilidad en el proceso de *parseo* del contenido. Además, al abstraer el acceso, se facilita el intercambio de información entre múltiples plataformas [25].

En la figura 11 se muestra el diseño de la arquitectura cliente-servidor del sistema. El navegador ejecuta la parte cliente de la aplicación, código HTML y JavaScript obtenido de la compilación de los distintos componentes Angular que forman la aplicación (ej.: plantillas, componentes, servicios, etc.). El cliente realiza peticiones a través de HTTP a los servicios que expone la API REST, la cual sin conocimiento del cliente, procesa la petición y recupera los datos necesarios de la base de datos. Finalmente, el cliente recibe la respuesta del servicio.

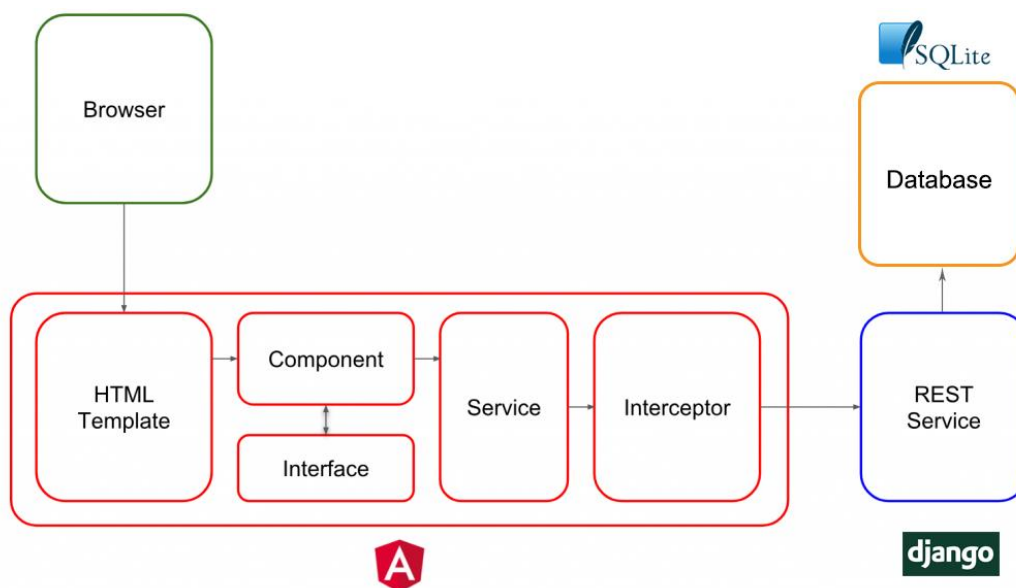


Figura 11: Esquema representativo de la arquitectura del sistema

6.1. Patrones de diseño

En este apartado se explicarán los patrones de diseño que siguen los *frameworks* utilizados en la implementación del sistema.

6.1.1. Capa de presentación

El patrón de diseño que sigue la capa de presentación no es el modelo-vista-controlador clásico, puesto que el modelo tiene una dependencia muy fuerte con la vista. Esto ocurre por uno de los conceptos base de Angular *two-way data binding* que hace que la forma de sincronizar los datos entre la vista y el modelo sea muy dependiente. Es decir, desde la vista se puede modificar el modelo y en el modelo es posible modificar la vista. Debido a esta característica, la independencia que se da en el patrón clásico de modelo-vista-controlador no se produce, y por lo tanto se suele decir que Angular sigue un patrón modelo-vista vista-modelo (MVVM) [26]. Existe un debate acerca de qué modelo determina la propiedad de *two-way data binding*. De hecho, también existe el modelo como lógica de negocio, como pueden ser los servicios o dependencias que se inyecten, que aunque se puede considerar como modelo está totalmente desacoplado de la vista.

6.1.2. Servicio REST

El cliente, a través peticiones HTTP, consume un servicio REST implementado con Django. La API, como se explicó anteriormente, se conecta con una base de datos relacional SQLite alojada en el mismo servidor que el servicio. Según el sitio oficial de Django [27], el *framework* tampoco sigue el patrón MVC clásico. Utiliza un patrón modelo-plantilla-vista (MTV). En Django una vista describe qué datos son presentados, y por tanto la vista delega en las plantillas (que son las que describen cómo se presenta el dato).

7. DISEÑO DE LA APLICACIÓN

En este capítulo se tratará el diseño y la implementación de las funcionalidades más significativas de la aplicación, comenzando por la aplicación web y terminando con la API implementada.

Los usuarios finales de esta aplicación engloban todo el espectro de profesores, con todo tipo de experiencia en aplicaciones informáticas. Por este motivo, el propósito ha sido hacer un diseño atractivo y simple para que el docente encuentre todas las funcionalidades a simple vista. Para lograrlo se han seguido las líneas de diseño Material Design [28], desarrolladas por Google.

Como se comentó en el capítulo tres que trataba sobre las tecnologías utilizadas, para la implementación de la capa de presentación se ha utilizado Angular. Entre las razones principales de utilizar este *framework* de desarrollo se encuentran:

- La posibilidad de reducir el número de peticiones al servidor gracias a la ejecución de parte del código en el lado del cliente.
- Poder escribir código modular, reutilizable y seguir los principios básicos del desarrollo de software.
- La facilidad de comunicar la capa de presentación con la API que se ha desarrollado a través de los mecanismos que ofrece Angular.

7.1. Colores y tipografía

Los colores y la tipografía utilizados han sido los de la documentación oficial del estilo web de la UCM [29]. Al tratarse de una aplicación cuyos potenciales usuarios finales son trabajadores de la UCM, se ha procurado ofrecer una sensación de continuidad entre la herramienta propuesta y el resto de las aplicaciones internas. En la figura 12 y 13 se muestran los códigos de color y tipografías utilizadas, respectivamente.

A screenshot of a code editor showing SCSS code for color variables. The editor has two tabs: 'variables.scss' and 'styles.scss'. The current file is 'variables.scss'. The code is as follows:

```
1 /*
2     Basado en los colores oficiales propuestos en el manual de estilo web de
3     la Universidad Complutense de Madrid
4
5     https://siii.ucm.es/colores-y-tipografia
6 */
7
8 $primary: #454545;
9 $accent: #990033;
10 $warn: $accent;
11 $links: #f8cd1e;
```

Figura 12: Fragmento de código - Códigos de color basados en el manual de estilo de la UCM

```
@import url("https://fonts.googleapis.com/icon?family=Material+Icons");
@import url('https://fonts.googleapis.com/css?family=Quattrocento+Sans');
@import './variables.scss';
@import './custom-style.scss';
@import './calendar.scss';
body {
  margin: 0;
  font-family: Quattrocento Sans, sans-serif;
  background-color: #ffffff;
}
```

Figura 13: Fragmento de código - Fuentes de texto utilizadas en la aplicación web

7.2. Funcionalidad de la aplicación

En este apartado se describen las funcionalidades implementadas y las pantallas asociadas más representativas de la aplicación.

En la figura 14 se muestra un diagrama de los módulos que componen la aplicación: *AppModule* (módulo raíz), *AsignaturasModule*, *AuthModule* (sistema de autenticación), *EleccionDocenciaModule*, *ProfesoresModule* y *UtilsComponentsModule* (módulo con clases de utilidad).

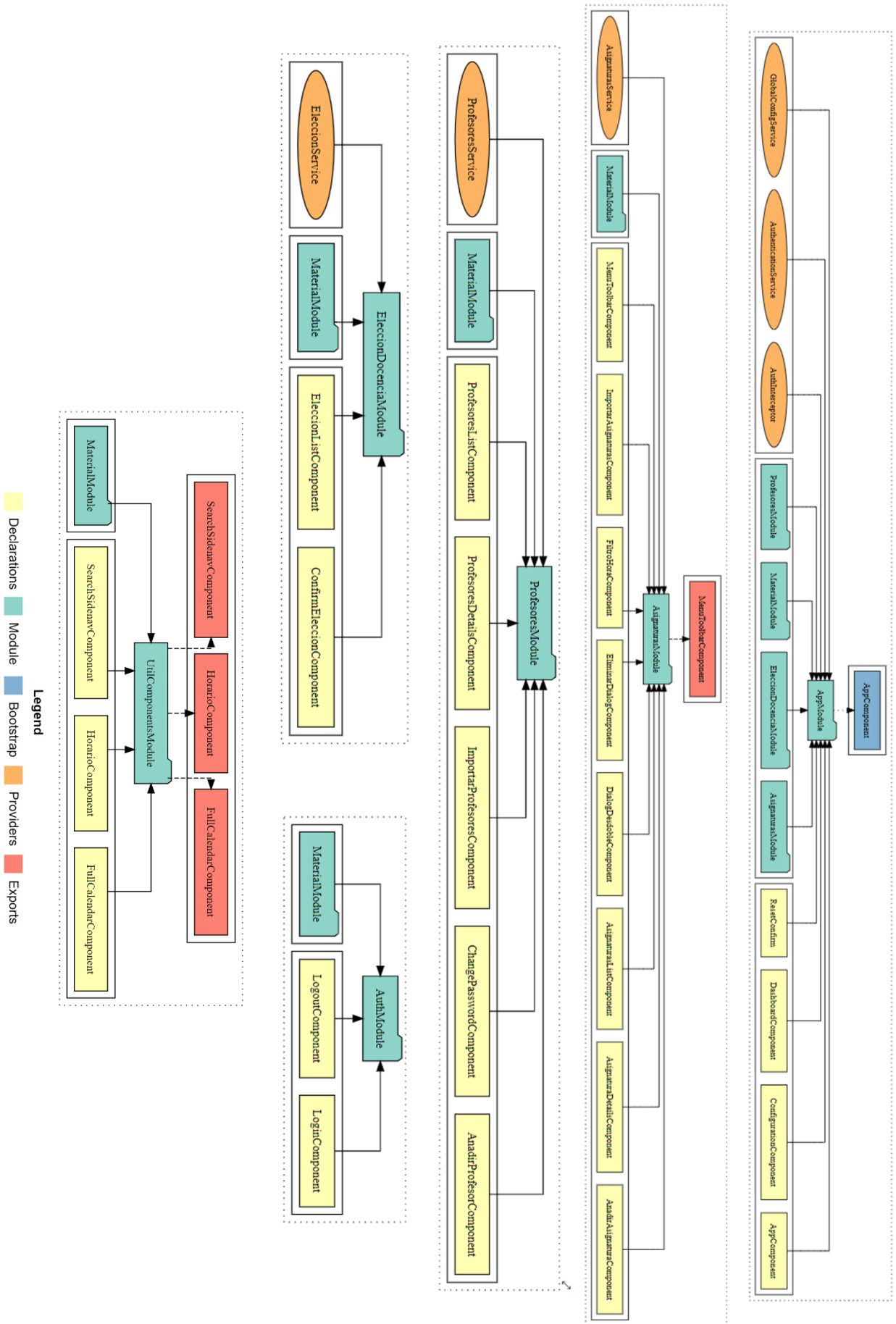


Figura 14: Diagrama de módulos de la aplicación web

7.2.1. Departamentos

7.2.1.1. Dashboard

El *dashboard* es la primera página a la que accede el usuario tras identificarse en el sistema. En esta página se muestran las principales métricas del proceso de elección de docencia que está en curso (ver figura 15).

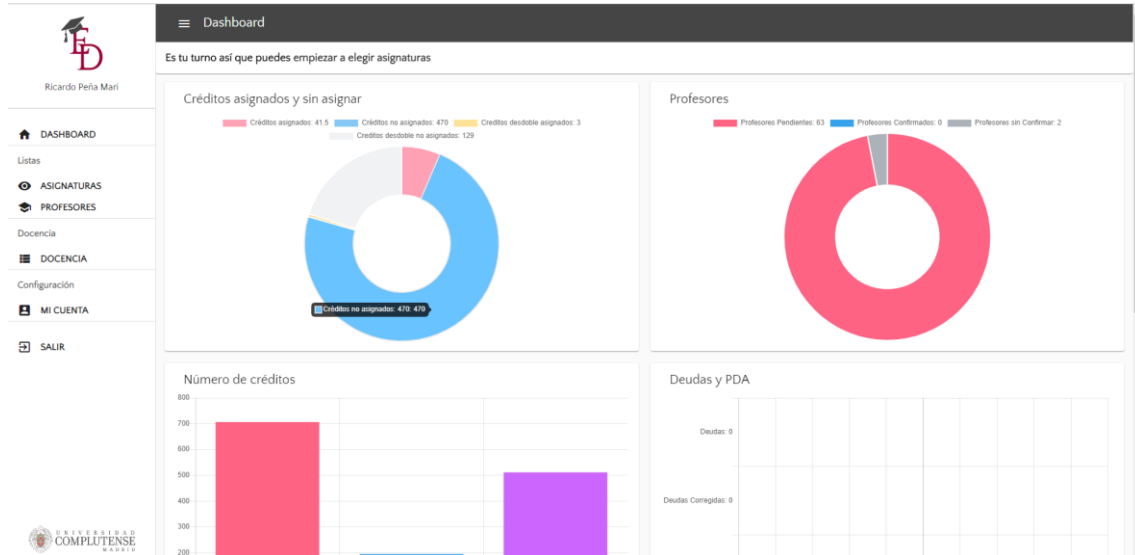


Figura 15: Pantalla del Dashboard

Las gráficas se han implementado utilizando la librería Chart.js adaptada para Angular [30]. En la figura 16 se puede ver un ejemplo del código HTML empleado y en la figura 17 la asignación de valores en el modelo.

```
14 <div class="grafics-container">
15 <mat-card class="grafica">
16 <mat-card-header>
17 <mat-card-title>Créditos asignados y sin asignar</mat-card-title>
18 </mat-card-header>
19 <mat-card-content>
20 <div style="display: block">
21 <canvas baseChart [datasets]="graficaCreditos.datas" [labels]="graficaCreditos.labels"
22 [chartType]="graficaCreditos.type" [options]="options"></canvas>
23 </div>
24 </mat-card-content>
25 </mat-card>
```

Figura 16: Fragmento de código - Gráfica de la pantalla de Dashboard HTML

```
this.graficaCreditos.datas = [
  { data: [
    this.departamento.creditos_asignados,
    this.departamento.creditos_sin_asignar,
    this.departamento.creditos_desdoble - this.departamento.creditos_desdobles_sin_asignar,
    this.departamento.creditos_desdobles_sin_asignar],
    label: "Créditos"
  }
];
this.graficaCreditos.labels = [
  'Créditos asignados: ' + this.departamento.creditos_asignados,
  'Créditos no asignados: ' + this.departamento.creditos_sin_asignar,
  'Creditos desdoble asignados: ' + (this.departamento.creditos_desdoble - this.departamento.creditos_desdobles_sin_asignar),
  'Creditos desdoble no asignados: ' + this.departamento.creditos_desdobles_sin_asignar
]
```

Figura 17: Fragmento de código - Asignación de valores a la gráfica del Dashboard

Observar que mediante la representación gráfica de las distintas métricas del sistema es posible monitorear el proceso de elección de docencia e identificar potenciales anomalías que se estén produciendo. Además, si se necesitan las métricas exactas se pueden ver de forma tabulada en la parte inferior de la página.

Por último, en la parte superior de esta vista se muestra un mensaje que indica si el usuario identificado le corresponde el turno de elección, o en caso de tener permisos de administración, se facilita un enlace a la configuración del sistema.

7.2.1.2. Configuración del departamento

La herramienta es capaz de extraer a partir del listado de asignaturas y profesores todas las variables necesarias para arbitrar el proceso de elección de docencia. Sin embargo, el administrador tiene la posibilidad de modificar manualmente las variables (ver figura 18) en caso de que una situación excepcional lo requiera. Así, por ejemplo, el administrador de la aplicación puede modificar la capacidad docente del departamento, la carga dinámica, el número total de deudas, etc.

Sin embargo, siempre será necesaria la intervención del administrador para modificar el inicio del proceso de elección y el reinicio de todas las elecciones docentes. Esta última funcionalidad elimina todo el rastro de las elecciones que existan en el sistema.

Departamento

*Ojo! Algun cambio puede desconfigurar el sistema, utilizar con precaucion

siglar *

SIC

Nombre del departamento *

Sistemas Informáticos y Computación

Total de créditos *

7065

Deudas *

0

Deudas Consegidas *

0

PDA *

0

Total créditos carga *

7065

Créditos fijos *

195

Resto de créditos *

5115

Créditos asignados *

415

Créditos sin asignar *

470

Capacidad examen *

4635

Carga dinámica *

1.0140237324703345

Eleccion de docencia iniciada

Guardar

Reiniciar docencia

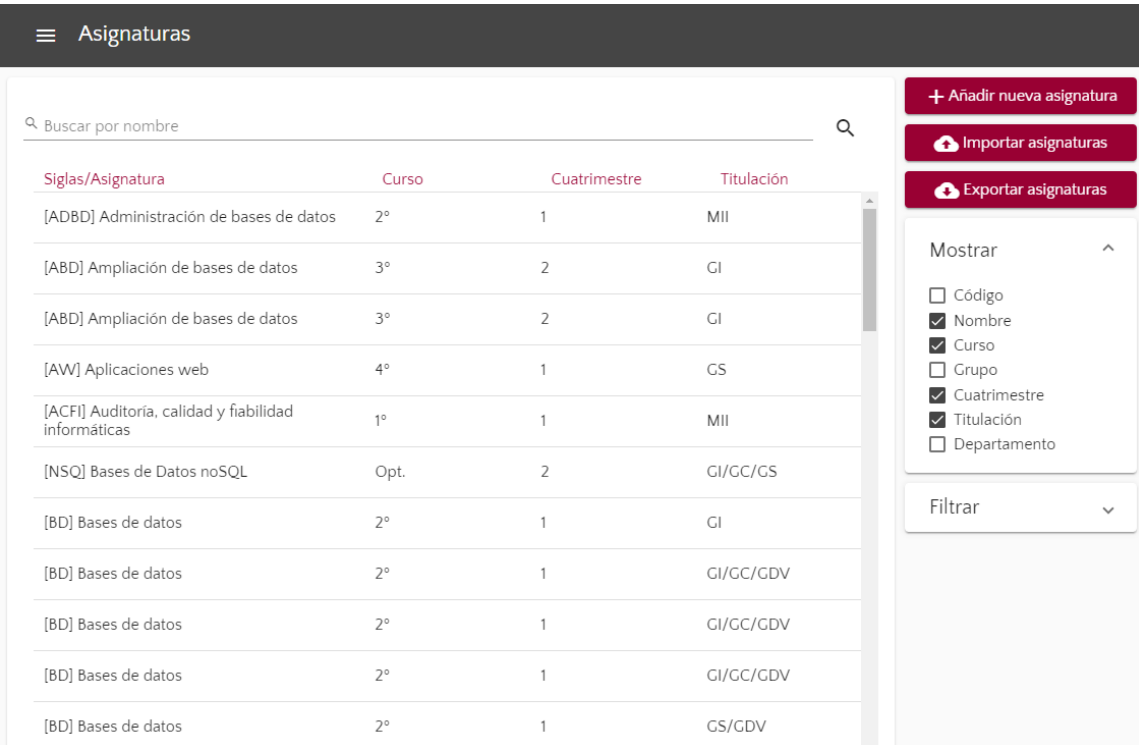
Figura 18: Página de configuración de la elección de docencia

7.2.2. Asignaturas

El módulo Asignaturas es uno de los pilares del almacenamiento y gestión de datos de la aplicación. Es el encargado de agrupar las funcionalidades que se explicarán a continuación.

7.2.2.1. Lista de asignaturas

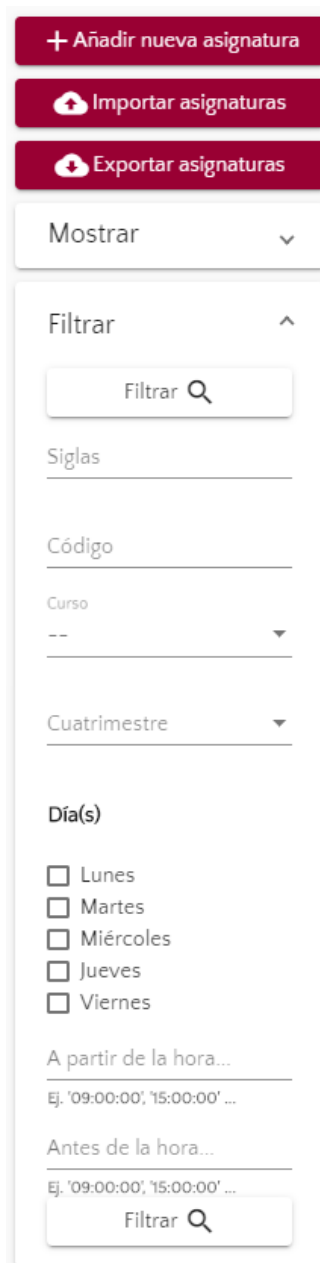
La lista de asignaturas está ordenada alfabéticamente por el nombre. Inicialmente se muestran cuatro columnas para no saturar de elementos la interfaz: las siglas y el nombre, el curso al que pertenece, el cuatrimestre y la titulación (ver figura 19).



Siglas/Asignatura	Curso	Cuatrimestre	Titulación
[ADBD] Administración de bases de datos	2º	1	MII
[ABD] Ampliación de bases de datos	3º	2	GI
[ABD] Ampliación de bases de datos	3º	2	GI
[AW] Aplicaciones web	4º	1	GS
[ACFI] Auditoría, calidad y fiabilidad informáticas	1º	1	MII
[NSQ] Bases de Datos noSQL	Opt.	2	GI/GC/GS
[BD] Bases de datos	2º	1	GI
[BD] Bases de datos	2º	1	GI/GC/GDV
[BD] Bases de datos	2º	1	GI/GC/GDV
[BD] Bases de datos	2º	1	GI/GC/GDV
[BD] Bases de datos	2º	1	GS/GDV

Figura 19: Listado de las asignaturas

Para facilitar la búsqueda de asignaturas, en la parte superior de la lista, se proporciona un campo de texto donde se puede buscar por el nombre de la asignatura. En la parte derecha se encuentra el panel extensible “Filtrar”. El componente es similar al de “Mostrar”, salvo que en este se muestra una serie de filtros con todas las opciones que tiene el usuario para filtrar las asignaturas (visible en la figura 20).



Panel de filtros de las asignaturas. Incluye botones para '+ Añadir nueva asignatura', 'Importar asignaturas' y 'Exportar asignaturas'. Un menú desplegable 'Mostrar' con una flecha hacia abajo. Una sección 'Filtrar' con una flecha hacia arriba, un campo de búsqueda 'Filtrar' con un icono de lupa, y campos de entrada para 'Siglas', 'Código' y 'Curso' (con un menú desplegable '--'). Una sección 'Cuatrimestre' con un menú desplegable. Una sección 'Día(s)' con casillas de verificación para 'Lunes', 'Martes', 'Miércoles', 'Jueves' y 'Viernes'. Sección de tiempo con 'A partir de la hora...' y 'Antes de la hora...' (ambas con ejemplos de hora y un icono de lupa).

Figura 20: Panel de filtros de las asignaturas

Cuando se hace uso de la búsqueda, ya sea por código de la asignatura, nombre o siglas, la lista se actualiza exactamente igual que con el resto de los tipos de búsqueda, salvo que en este caso se resalta de color amarillo el texto coincidente con el buscado (ver figura 21). Para lograr este efecto se recorren todos los resultados de la búsqueda, se comparan con una expresión regular, y se modifica el código HTML para conseguir resaltar la información tal como se muestra en la figura 22.

Buscar por nombre

Ampl

Código	Siglas/Asignatura
803284	[ABD]Ampliación de bases de datos
803284	[ABD]Ampliación de bases de datos

Figura 21: Búsqueda de asignaturas por texto

```
highlightResults(): void {
  this.asignaturas.forEach(asignatura => {
    if (asignatura.nombre && this.searchVals.nombre) {
      var regex = new RegExp(this.searchVals.nombre, 'gi')
      asignatura.nombre = asignatura.nombre.replace(regex, function (str) { return "<span class='highlight'>" + str + "</span>" });
    };
    if (asignatura.siglas && this.searchVals.siglas) {
      var regex = new RegExp(this.searchVals.siglas, 'gi')
      asignatura.siglas = asignatura.siglas.replace(regex, function (str) { return "<span class='highlight'>" + str + "</span>" });
    };
    if (asignatura.codigo && this.searchVals.codigo) {
      var regex = new RegExp(this.searchVals.codigo, 'gi')
      asignatura.codigo = asignatura.codigo.replace(regex, function (str) { return "<span class='highlight'>" + str + "</span>" });
    };
  });
}
```

Figura 22: Fragmento de código - Resaltar las asignaturas filtradas

La obtención de resultados a partir de los parámetros de búsqueda se hace a nivel de API, por lo que la capa de presentación sólo observa los cambios en los formularios, manda la petición y actualiza la lista con los resultados obtenidos. En la figura 23 se muestra la función que filtra las asignaturas y hace la llamada a la API.

```
src > app > asignaturas > asignaturas-list > asignaturas-list.component.ts > AsignaturasListComponent
106 else {
107   this.searchVals.dias.splice(index, 1);
108 }
109 }
110 search(): void {
111   this.loading = true;
112   var diasAux = []
113   if (this.searchVals.dias.length < 5) {
114     diasAux = this.searchVals.dias;
115   }
116   this.asignaturasService.searchAsignatura(
117     this.searchVals.siglas,
118     this.searchVals.nombre,
119     this.searchVals.codigo,
120     this.searchVals.curso,
121     this.searchVals.cuatrimestre,
122     this.searchVals.ini,
123     this.searchVals.fin,
124     diasAux
125   ).subscribe(asignaturas => {
126     this.asignaturas = asignaturas;
127     this.loading = false;
128     this.highlightResults();
129   });
130 }
131 highlightResults(): void {
132   this.asignaturas.forEach(asignatura => {
133     if (asignatura.nombre && this.searchVals.nombre) {
134       var regex = new RegExp(this.searchVals.nombre, 'gi')
135       asignatura.nombre = asignatura.nombre.replace(regex,
136     );
137   });
138 }

src > app > services > asignaturas.service.ts > AsignaturasService > saveAsignatura > subscribe()
37 getAllCalendarios(): Observable<Calendario[]> {
38   return this.http.get<Calendario[]>(this.calendariosUrl);
39 }
40 }
41 searchAsignatura(
42   siglas: string,
43   nombre: string,
44   codigo: string,
45   curso: string,
46   cuatrimestre: number,
47   inicio: string,
48   fin: string,
49   dia: string[]
50 ): Observable<Asignatura[]> {
51   var params = 'siglas=' + siglas +
52     'nombre=' + nombre +
53     'codigo=' + codigo +
54     'curso=' + curso +
55     'inicio=' + inicio +
56     'fin=' + fin;
57   if (cuatrimestre) {
58     params += 'cuatrimestre=' + cuatrimestre
59   }
60   dia.forEach(d => {
61     params += 'dia=' + d
62   });
63   return this.http.get<Asignatura[]>
64     (this.asignaturasUrl + '?' + params);
65 }
66 saveAsignatura(asignatura: Asignatura): void {
67   if (asignatura.dia.length > 0) {
68     this.dia.forEach(d => {
69       asignatura.dia.push(d);
70     });
71   }
72   return this.http.post<Asignatura>(this.asignaturasUrl, asignatura);
73 }
74 }
75 }
```

Figura 23: Fragmento de código - Búsqueda de asignaturas

El último elemento son los botones de acción, estos solo están disponibles para los administradores y realizan las funcionalidades de añadir, importar y exportar asignaturas. El primero y el segundo botón se describirán más adelante, puesto que

conducen a otros componentes. El tercer botón permite exportar las asignaturas a una hoja de cálculo con el mismo formato que se requiere para importar las asignaturas.

7.2.2.2. Detalle de la asignatura

En primer lugar, se ha decidido incluir una miga de pan (del inglés *breadcrumb*) [31]. Consiste en una línea de texto en la que se indica el recorrido seguido y la manera de retornar. Permite que el usuario conozca la ruta de su ubicación y navegue a través de ella.

Los detalles de la asignatura se han dividido en pestañas con el fin de no saturar de información al usuario y organizar la vista de una manera clara y efectiva (ver figura 24). En la segunda pestaña, como se puede ver en la figura 25, está el horario en el que se desarrolla la asignatura, se muestra de dos maneras diferentes, en texto plano y en formato de calendario semanal.

The screenshot shows the 'asignaturas / ABD' interface. The breadcrumb is '[ABD] Ampliación de bases de datos'. The 'Horario' tab is active. The details are as follows:

Código	803284
Cuatrimestre	2 cuatrimestre
Curso	3ºB
Calendario	Grados y máster (2019-01-29 - 2019-05-17)
Departamento	SIC
Titulación	GI
Créditos	6
Docencia teoría	David de Frutos Escrig
Docencia desdoble	Sin docencia

Figura 24: Detalle de la asignatura, primera pestaña

The screenshot shows the 'Horario' tab. It displays the schedule in two formats: a text summary and a weekly calendar grid.

Text Summary:

- Lunes: 09:00:00 - 09:50:00 [Aula 13]
- Miércoles: 09:00:00 - 09:50:00 [Aula 13]
- Martes: 09:00:00 - 10:50:00 [Aula 13]
- Martes: 09:00:00 - 10:50:00 [Lab. 7, 8]

Weekly Calendar Grid:

	Lunes	Martes	Miércoles	Jueves	Viernes
09:00:00	Aula 13	Aula 13	Lab. 7, 8	Aula 13	
10:00:00		Aula 13	Lab. 7, 8		
11:00:00					
12:00:00					
13:00:00					
14:00:00					
15:00:00					
16:00:00					
17:00:00					
18:00:00					
19:00:00					
20:00:00					

* desdobles

Figura 25: Detalle de la asignatura, segunda pestaña

Por último, los botones de la parte superior derecha solo son accesibles para los administradores y permiten editar o eliminar la asignatura. En caso de eliminarla se le mostrará al usuario un *pop-up* (ver figura 26). Una vez confirmada la acción se le enviará la petición a la API. Si lo que se desea es editarla, se conducirá al usuario al componente que se expondrá a continuación.

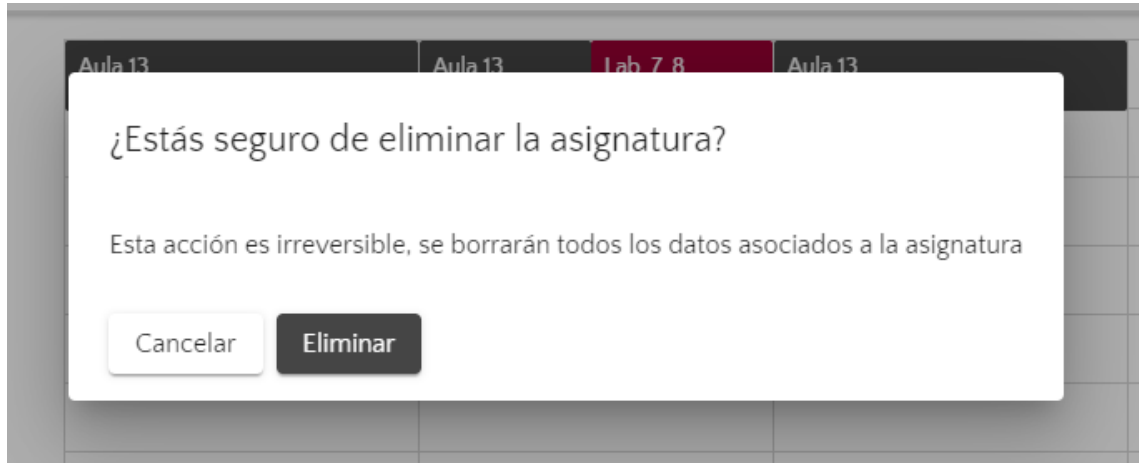


Figura 26: Confirmación para borrar asignatura

7.2.2.3. Añadir o editar asignatura

El componente utilizado para añadir una añadir o editar una asignatura nueva es el mismo. La única diferencia entre ambos es que en el primer caso los campos aparecen vacíos, y en el segundo completados con los datos de la asignatura.

El formulario es guiado, de manera que el usuario no se encuentra toda la información en una sola vista, si no que conforme acaba un paso, avanza al siguiente, siempre teniendo la posibilidad de retroceder. En las figuras 27, 28, 29 y 30 se muestran los pasos del formulario.

A screenshot of a mobile application form titled "Añadir asignatura". At the top, there is a progress bar with five steps: "1 Detalles generales", "2 Departamento", "3 Desdobles", "4 Horarios", and "5 Guardar". The first step is active. Below the progress bar, there are several input fields with labels and asterisks indicating they are required: "Nombre *", "Siglas *", "Código *", "Cuatrimestre *", "Curso *", "Grupo *", "Calendario *", and "Titulación *". The values entered in these fields are: "Administración de bases de datos", "ADBD", "607300", "1 cuatrimestre", "2º", "A", "Grados y máster (2018-09-12 - 2018-12-20)", and "MII". At the bottom of the form, there is a "Siguiente" button.

Figura 27: Editar asignatura, paso uno

← Añadir asignatura

1 Detalles generales 2 Departamento 3 Desdobles 4 Horarios 5 Guardar

Siglas del departamento *

SIC

Atrás Siguiete

Figura 28: Editar asignatura, paso dos

← Añadir asignatura

1 Detalles generales 2 Departamento 3 Desdobles 4 Horarios 5 Guardar

Añadir desdoble

Desdoble #1

Añadir horario al desdoble

Horario #1

Aula/Laboratorio *

Lab. 7, 8

Ej. 'Aula 1', 'Aula 13', 'Laboratorio 4' ...

Día de la semana *

M

Ej. 'L', 'M', 'X', 'J', '...' ...

Hora de inicio *

09:00:00

Ej. '09:00:00', '15:00:00' ...

Hora de fin *

10:50:00

Ej. '09:00:00', '15:00:00' ...

Atrás Siguiete

Figura 29: Editar asignatura, paso tres

← Añadir asignatura

1 Detalles generales 2 Departamento 3 Desdobles 4 Horarios 5 Guardar

Añadir horario

Horario 1

Aula/Laboratorio *

Aula 13

Ej. 'Aula 1', 'Aula 13', 'Laboratorio 4' ...

Día de la semana *

L

Ej. 'L', 'M', 'X', 'J', '...' ...

Hora de inicio *

09:00:00

Ej. '09:00:00', '15:00:00' ...

Hora de fin *

09:50:00

Ej. '09:00:00', '15:00:00' ...

Atrás Siguiete

Figura 30: Editar asignatura, paso cuatro

← Añadir asignatura

1 Detalles generales 2 Departamento 3 Desdobles 4 Horarios 5 Guardar

Ya puedes salvar los cambios

Atrás Volver a empezar Guardar

Figura 31: Editar asignatura, paso cinco

En el paso final (ver figura 31), al usuario se le ofrecen tres posibilidades, retroceder, volver al inicio o guardar los cambios. Al pulsar sobre la opción de guardar se realizará la comprobación de que todos los campos estén completos. En caso

afirmativo se mandará la petición a la API para que persista en la base de datos (ver figura 32). Si hubiera algún error, se le notificaría al usuario con un mensaje informativo.

```
saveAsignatura(asignatura: Asignatura): void {  
  
  if (asignatura.id !== undefined) {  
    this.http.put<Asignatura>(this.asignaturasUrl + asignatura.id + '/', asignatura)  
      .subscribe(  
        data => { // data is already a JSON object  
          this.router.navigate(['asignaturas/' + asignatura.id]);  
          this.avisosService.enviarMensaje("Se han actualizado los cambios correctamente");  
        },  
        err => {  
          this.avisosService.enviarMensaje("Error al editar la asignatura, por favor, revisa el formulario");  
        }  
      );  
  }  
  else {  
    this.http.post<Asignatura>(this.asignaturasUrl, asignatura)  
      .subscribe(data => { // data is already a JSON object  
        this.router.navigate(['asignaturas/' + data.id]);  
        this.avisosService.enviarMensaje("Se ha creado la asignatura correctamente");  
      },  
      err => {  
        this.avisosService.enviarMensaje("Error al crear la asignatura, por favor, revisa el formulario");  
      });  
  }  
}
```

Figura 32: Fragmento de código - Función para añadir/editar asignatura

7.2.2.4. Importar asignaturas

Como se ha comentado anteriormente, a esta funcionalidad se accede a través de la lista de asignaturas. En esta vista se muestra una breve aclaración del formato que debe tener la lista de asignaturas que se desea importar. Se proporcionan tres campos de texto y un *checkbox* que se enviarán junto al archivo, el cual se adjuntará pulsando sobre el botón que permite seleccionar un fichero (ver figura 33). En la figura 34 se muestra un fragmento del código HTML de esta página.

Finalmente, cuando el usuario pulsa sobre el botón de importar se envía una petición al servidor. Este se encargará del procesamiento y control de errores de los datos, en caso de detectarlos responderá con un código de error y se le notificará al usuario.

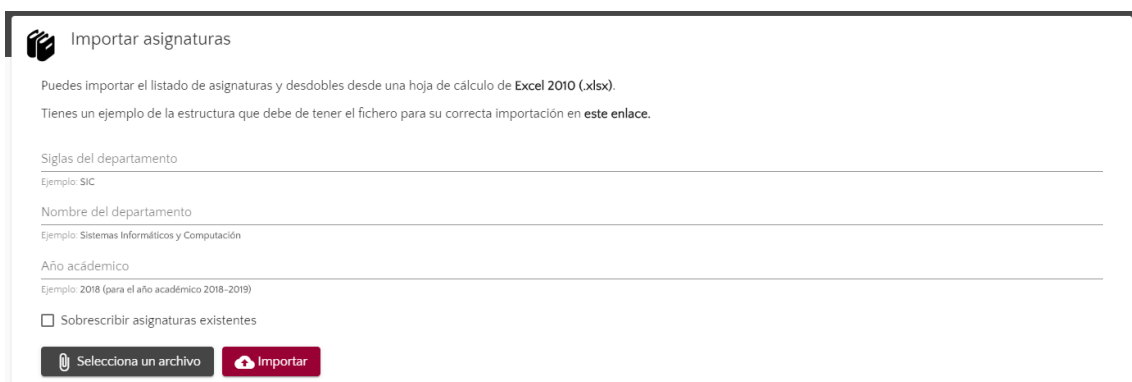


Figura 33: Página para importar asignaturas

```

importar-asignaturas.component.html x
src > app > asignaturas > importar-asignaturas > importar-asignaturas.component.html > div.form-container
1 <div class="decorator"></div>
2 <div class="form-container">
3 <mat-card class="card">
4 <mat-card-header>
5 <div mat-card-avatar class="header-image"></div>
6 <mat-card-title>Importar asignaturas</mat-card-title>
7 </mat-card-header>
8 <mat-card-content class="formulario">
9 <p>Puedes importar el listado de asignaturas y desdobles desde una hoja de cálculo de <strong>Excel 2010
10 <strong>(xlsx)</strong></p>
11 <p>Tienes un ejemplo de la estructura que debe de tener el fichero para su correcta importación en
12 <strong>este enlace.</strong></p>
13
14 <div class="importar-formulario">
15 <mat-form-field>
16 <input matInput [(ngModel)]="departamento_siglas" placeholder="Siglas del departamento">
17 <mat-hint>Ejemplo: <strong>SIC</strong></mat-hint>
18 </mat-form-field>
19
20 <mat-form-field>
21 <input matInput [(ngModel)]="departamento_nombre" placeholder="Nombre del departamento">
22 <mat-hint>Ejemplo: <strong>Sistemas Informáticos y Computación</strong></mat-hint>
23 </mat-form-field>
24
25 <mat-form-field>
26 <input matInput [(ngModel)]="anyo" placeholder="Año académico">
27 <mat-hint>Ejemplo: <strong>2018 (para el año académico 2018-2019)</strong></mat-hint>
28 </mat-form-field>
29 <br>
30 <mat-checkbox [(ngModel)]="sobrescribir">Sobrescribir asignaturas existentes</mat-checkbox>
31 </div>
32
33 <input style="display: none" type="file" (change)="onFileChanged($event)" #fileInput>
34
35 <button mat-raised-button color="primary" (click)="fileInput.click()">
36 <mat-icon>attach_file</mat-icon> Selecciona un archivo
37 </button>
38
39 <button mat-raised-button color="accent" (click)="importar()">
40 <mat-icon>cloud_upload</mat-icon> Importar
41 </button>
42
43 <div class="carga" *ngIf="progreso">
44 <mat-spinner [diameter]="30"></mat-spinner>
45 <p>Importando asignaturas ... </p>
46 </div>
47
48 </mat-card-content>
49 </mat-card>
50 </div>

```

Figura 34: Fragmento de código - HTML de la página para importar asignaturas

7.2.3. Profesores

El módulo Profesores es muy similar al de Asignaturas, también incluye un listado, un formulario para añadir o editar, una vista de detalles y la opción de importar profesores al sistema. A continuación, se comentarán las diferentes funcionalidades haciendo hincapié en las diferencias encontradas con el anterior módulo.

7.2.3.1. Lista de profesores

Para representar el listado de profesores se ha tratado de respetar la misma estructura y estilo que el listado de asignaturas.

En el caso de la lista de profesores, se ha incorporado una columna de docencia que muestra el estado de la elección docente mediante un icono, que, a través de diferentes colores, indica si el profesor ha realizado la elección y si ésta ha sido validada por un administrador (ver figura 35).

Profesores					
Buscar por nombre			Buscar por apellidos		
Docencia	Escalafon	Apellidos	Nombre	Email	Categoría
<input checked="" type="checkbox"/>	1	Frutos Escrig	David de	fake.defrutos@sip.ucm.es.fake	CU
<input checked="" type="checkbox"/>	2	Peña Mari	Ricardo	fake.ricardo@sip.ucm.es.fake	CU
	3	Martí Oliet	Narciso	fake.narciso@sip.ucm.es.fake	CU
	4	López Fraguas	Francisco J.	fake.fraguas@sip.ucm.es.fake	CU
	5	Núñez García	Manuel	fake.mn@sip.ucm.es.fake	CU
	6	Fernández Camacho	Maria Inés	fake.minesfc@sip.ucm.es.fake	TU
	7	Bradley Delso	Margarita	fake.bradley@sip.ucm.es.fake	TU
	8	Gil Luezas	Ana	fake.anagil@sip.ucm.es.fake	TU
	9	Ortega Mallén	Yolanda	fake.yolanda@sip.ucm.es.fake	TU
	10	Cavilanes Franco	Antonio	fake.agav@sip.ucm.es.fake	TU
	11	Nieva Soto	Susana	fake.nieva@sip.ucm.es.fake	TU
	12	Llana Díaz	Luis Fernando	fake.llana@sip.ucm.es.fake	TU
	13	Arenas Sánchez	Purificación	fake.puri@sip.ucm.es.fake	TU
	14	Pareja Flores	Cristóbal	fake.cpareja@sip.ucm.es.fake	CEU
	15	Martin de la Calle	Pedro J.	fake.pjmartin@sip.ucm.es.fake	TU

Figura 35: Listado de profesores

Las columnas que se muestran por defecto serán: el estado de la docencia, el escalafón, los apellido, el nombre, el email y la categoría a la que pertenece el profesor. En la figura 36 se puede ver el fragmento de código que las define, a la izquierda el modelo y a la derecha el HTML que las maqueta.

```

profesores-list.component.ts
30  apellidos: boolean,
31  nombre: boolean,
32  email: boolean,
33  telefono: boolean,
34  despacho: boolean,
35  departamento: boolean,
36  categoria: boolean,
37  };
38  numCols: number;
39  searchVals: {
40    nombre: string,
41    apellidos: string,
42    email: string,
43    telefono: string,
44    despacho: number,
45    escalafon: number,
46    categoria: string
47  };
48  loading: boolean;
49  categorias: Categoria[];
50
51
52  constructor(private profesoresService: ProfesoresService, private router: Router, private title
53    this.titleService.setTitle('Profesores');
54    this.admin = this.globalConfigService.isAdmin();
55    this.selected = 1;
56  };
57  this.opts = {
58    docencia: true,
59    escalafon: true,
60    nombre: true,
61    apellidos: true,
62    email: true,
63    telefono: false,
64    despacho: false,
65    departamento: false,
66    categoria: true,
67  };
68  this.searchVals = {
69    nombre: '',
70    apellidos: '',
71    email: '',
72    telefono: '',
73    despacho: undefined,
74    escalafon: undefined,
75    categoria: ''
76  };
77  };
78  };
79
80  ngOnInit() {
81    newToolbarComponent.updateTitle('Profesores');
82    this.loading = true;
83    this.numCols = 8;
84    this.getProfesores();
85    this.profesoresService.getCategorias()
86      .subscribe()
87      categorias => {
88        this.categorias = categorias;
89      }
90    }
91
profesores-list.component.html
23  </button>
24  </form>
25  </div>
26  <div class="total-container">
27  <div class="lista">
28  <div class="content">
29  <mat-grid-list cols="{numCols}" rowHeight="30px">
30  <mat-grid-tile *ngIf="opts.docencia">
31  <button mat-button color="accent" class="content-row">
32  Docencia
33  </button>
34  </mat-grid-tile>
35  <mat-grid-tile *ngIf="opts.escalafon">
36  <button mat-button color="accent" class="content-row">
37  Escalafon
38  </button>
39  </mat-grid-tile>
40  <mat-grid-tile *ngIf="opts.apellidos">
41  <button mat-button color="accent" class="content-row">
42  Apellidos
43  </button>
44  </mat-grid-tile>
45  <mat-grid-tile *ngIf="opts.nombre">
46  <button mat-button color="accent" class="content-row">
47  Nombre
48  </button>
49  </mat-grid-tile>
50  <mat-grid-tile *ngIf="opts.email">
51  <button mat-button color="accent" class="content-row nombre">
52  Email
53  </button>
54  </mat-grid-tile>
55  <mat-grid-tile *ngIf="opts.telefono">
56  <button mat-button color="accent" class="content-row">
57  Telefono
58  </button>
59  </mat-grid-tile>
60  <mat-grid-tile *ngIf="opts.despacho">
61  <button mat-button color="accent" class="content-row">
62  Despacho
63  </button>
64  </mat-grid-tile>
65  <mat-grid-tile *ngIf="opts.departamento">
66  <button mat-button color="accent" class="content-row">
67  Departamento
68  </button>
69  </mat-grid-tile>
70  <mat-grid-tile *ngIf="opts.categoria">
71  <button mat-button color="accent" class="content-row">
72  Categoria
73  </button>
74  </mat-grid-tile>

```

Figura 36: Fragmento de código - Definición y vista de las columnas del listado de profesores

En el listado de profesores, como en el de asignaturas, se ha incorporado un sistema de filtrado (ver figura 37). Es posible buscar profesores a través de su nombre, correo electrónico, teléfono, despacho, posición en el escalafón y puesto de trabajo.

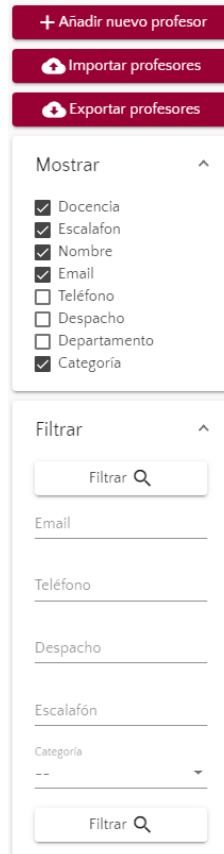


Figura 37: Filtros del listado de profesores

Con respecto a la búsqueda por texto, también se incluye, como en el anterior módulo, el resaltado de las coincidencias encontradas durante el filtrado (ver figura 38).

Docencia	Escalafon	Apellidos	Nombre	Email	Categoria
41		Rabanal Basalo	Pablo	fake.prabanal@fdi.ucm.es.fake	CD

Figura 38: Búsqueda de profesores por texto

Al pulsar sobre cualquier profesor de la lista se accede a su página de detalle, en el siguiente apartado se hablará de ella.

Al igual que ocurre con el listado de asignaturas, es posible exportar el listado de profesores en formato “.xlsx”. El documento que se exporta tiene la distribución de columnas que debería tener la hoja de cálculo que se utiliza durante la importación de profesores al sistema.

7.2.3.2. Detalle del profesor

La página de detalle de un profesor muestra al usuario todos los detalles de este, al igual que el detalle de la asignatura, consta de una miga de pan para retroceder al listado de asignaturas.

La información está dividida en dos pestañas, la primera, muestra los datos personales del profesor, y la cantidad de deuda y PDA que posee. En la segunda, se muestra la elección de docencia que lleva hasta el momento. En las figuras 39 y 40 se puede ver el ejemplo de un docente que ya ha incluido asignaturas en su elección.

profesores / David de Frutos Escrig, David de				
Eliminar Editar				
Detalles generales		Docencia		
Nombre	David de			
Apellidos	Frutos Escrig			
Email	defrutos@sip.ucm.es			
Deuda	2014: 0 créditos	2015: 0 créditos	2016: 0 créditos	2017: 0 créditos
Teléfono	4529			
Despacho	310C(Mat)			
PDA	0 créditos			
Categoría	Catedrático de Universidad			
Departamento	SIC			

Figura 39: Detalle del profesor, primera pestaña

profesores / David de
Frutos Escrig, David de

Eliminar Editar

Detalles generales Docencia

Ver en el gestor

[ABD] Ampliación de bases de datos
Horario:
L: 09:00:00-09:50:00 (Aula 13)
X: 09:00:00-09:50:00 (Aula 13)
M: 09:00:00-10:50:00 (Aula 13)
2 cuatrimestre - 3ºB
Teoría + desdoble
DETALLES

[EDA] Estructura de datos y algoritmos
Horario:
L: 14:00:00-15:50:00 (Aula 10)
M: 14:00:00-14:50:00 (Aula 10)
1 cuatrimestre - 2ºC
Teoría + desdoble
DETALLES

[EDA] Estructura de datos y algoritmos
Horario:
M: 15:00:00-16:50:00 (Aula Mult.-1208)
J: 15:00:00-15:50:00 (Aula Mult.-1208)
1 cuatrimestre - 2ºF
Teoría + desdoble
DETALLES

[FP] Fundamentos de la Programación
Horario:
L: 11:00:00-11:50:00 (Aula 2)
J: 11:00:00-11:50:00 (Aula 2)
V: 11:00:00-12:50:00 (Aula 2)
2 cuatrimestre - 1ºB
Teoría + desdoble
DETALLES

[AW] Aplicaciones web

[NSQ] Bases de Datos noSQL
Horario:

Figura 40: Detalle del profesor, segunda pestaña

En la segunda pestaña, el usuario puede ver las asignaturas que ha elegido el profesor seleccionado del listado de profesores. Además, en caso de estar viendo su propio perfil o ser administrador, tiene la posibilidad de acceder al gestor de docencia, donde editar, eliminar o confirmar la elección.

Existen tres tipos de tarjetas a la hora de mostrar estas asignaturas, en las figuras 41, 42 y 43, se pueden encontrar la de una asignatura normal, un desdoble y una asignatura divisible, respectivamente. En caso de pulsar sobre el botón "Detalles", el usuario será redirigido a la página de detalle de la asignatura.

[ABD] Ampliación de bases de datos
Horario:
L: 09:00:00-09:50:00 (Aula 13)
X: 09:00:00-09:50:00 (Aula 13)
M: 09:00:00-10:50:00 (Aula 13)
2 cuatrimestre - 3ºB

Teoría + desdoble

DETALLES

Figura 41: Detalle profesor - Tarjeta asignatura normal

[AW] Aplicaciones web
Horario:
J: 16:00:00-17:50:00 (Lab. 9, 11)
1 cuatrimestre - 4ºE

Desdoble

DETALLES

Figura 42: Detalle profesor - Tarjeta desdoble

[NSQ] Bases de Datos noSQL
Horario:
L: 14:00:00-15:50:00 (Aula 9)
M: 14:00:00-15:50:00 (Aula 9)
2 cuatrimestre - Opt.A

Divisible

3 crédito(s)

DETALLES

Figura 43: Detalle profesor - Tarjeta asignatura divisible

7.2.3.3. Añadir o editar profesores

Esta funcionalidad está restringida a los usuarios que tengan permisos de administración. Tanto para añadir o editar se reutiliza la misma vista, salvo en el caso de editar un profesor donde el formulario tendrá los campos completados y solo será necesario modificar los que se deseen (ver figuras 44 y 45). El diseño de la vista es el mismo que se ha descrito en el apartado 7.2.2.3 para la creación y edición de asignaturas.

También, en la edición del profesor es posible dar permisos de administración a un profesor.

The screenshot shows a mobile application interface for editing a professor. At the top, there is a title bar with a back arrow and the text 'Editar profesor'. Below the title bar, there are three tabs: '1 Detalles generales' (active), '2 Deuda & PDA', and 'Guardar'. The 'Detalles generales' tab contains the following fields:

- Administrador
- Nombre *
David de
- Apellidos *
Frutos Escrig
- Despacho *
310C(Mat)
- Email *
defrutos@sip.ucm.es
- Teléfono *
4529
- Escalafón *
1
- Categoría
[CU] Catedrático de Universidad
- Departamento *
SIC

At the bottom of the form, there is a 'Siguiete' button.

Figura 44: Editar profesor, paso uno

Figura 45: Editar profesor, paso dos

Figura 46: Editar profesor, paso tres

En el paso tres, en caso de que el usuario quiera guardar los cambios que ha realizado, el sistema realizará una petición a la API (figura 46 y 47).

```

saveProfesor(profesor: Profesor): void {
  if (profesor.usuario.id !== undefined) {
    this.http.patch<Profesor>(this.profesoresUrl + profesor.usuario.id + '/', profesor)
      .subscribe(
        data => {
          this.router.navigate(['./profesores/' + profesor.usuario.id]);
          this.avisosService.enviarMensaje("Se han actualizado los cambios correctamente");
        },
        err => {
          this.avisosService.enviarMensaje("Error al editar el profesor, por favor, revisa los campos");
        }
      );
  }
  else {
    this.http.post<Profesor>(this.profesoresUrl, profesor)
      .subscribe(
        data => {
          this.router.navigate(['./profesores/' + data.usuario.id]);
          this.avisosService.enviarMensaje("Se ha creado la profesor correctamente");
        },
        err => {
          this.avisosService.enviarMensaje("Error al crear el profesor, por favor, revisa los campos");
        }
      );
  }
}

```

Figura 47: Fragmento de código - Función para añadir o editar profesor

Además, si se está creando un nuevo profesor se generará una contraseña segura y se le enviará por correo electrónico. Este envío se realizará desde el servidor que aloja la API. En la figura 48 se puede ver la función que genera la contraseña.

```
passwordGenerator() {
  var length = Math.floor((Math.random() * 18) + 10);
  var text = "";
  var possible = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+-?;/#@!;/*-+.";

  for (var i = 0; i < length; i++)
    text += possible.charAt(Math.floor(Math.random() * possible.length));

  return text;
}
```

Figura 48: Fragmento de código - Generador de contraseñas seguras

7.2.3.4. Importar profesores

La página de importar profesores es accesible desde el listado como se ha comentado anteriormente. En la figura 49 se puede ver la explicación de las condiciones para importar el archivo. El campo de texto situado inmediatamente después sirve para indicar a qué departamento pertenecen las asignaturas que se quieren importar. A continuación, se puede ver un *checkbox* que permite decidir si añadir o sobrescribir los profesores existentes, y un botón con el título “Selecciona un archivo” desde el que adjuntar el documento.

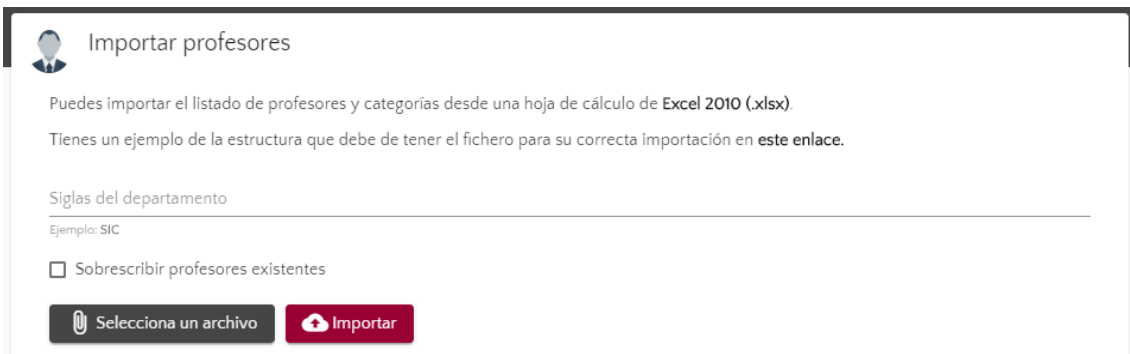


Figura 49: Página de importar profesores

En caso de que haya algún error durante la importación, se le mostrará al usuario un mensaje informativo. (Figura 50).

```
importar(archivo: File, departamento_siglas: string, sobrescribir: boolean): Observable<ProfesorImportar> {
  const uploadData = new FormData();
  uploadData.append('excel_file', archivo, archivo.name);
  uploadData.append('departamento_siglas', departamento_siglas);
  uploadData.append('sobrescribir', sobrescribir ? "true" : "false");

  return this.http.post<ProfesorImportar>(this.profesoresUrl + 'importar/', uploadData);
}
```

Figura 50: Fragmento de código - Petición para importar profesores

7.2.4. Elección de docencia

El módulo que se trata en esta sección consta de una única vista. Sin embargo, engloba la funcionalidad principal de la aplicación. En ella un profesor podrá elegir las

asignaturas a impartir al año siguiente. Con el fin de explicar cada detalle, se dividirá el módulo en tres partes bien diferenciadas. En primer lugar, el listado de asignaturas y los filtros, en segundo lugar, el panel de información y mensajes, a continuación el calendario, y, por último la funcionalidad de guardar y confirmar la docencia. Para evitar que el usuario se vea saturado con excesiva información, el menú lateral se minimiza automáticamente cuando accede a la página.

En la figura 51 se muestra esta vista con todos sus componentes visibles. Sin embargo, si el usuario que está eligiendo asignaturas cree necesitar más espacio, tiene la posibilidad de minimizar el menú y ocultar las distintas columnas que conforman esta vista (ver figura 52 y 53).

The screenshot displays the 'Elección Docencia' interface. At the top, there is a navigation bar with a hamburger menu, the title 'Elección Docencia', and a search filter 'Filtrar asignaturas'. Below this, the user's name 'David de Frutos Esc...' is shown. The main content area is divided into several sections:

- Información:** A confirmation message 'Esta elección está confirmada!' with a checkmark icon. It also shows 'Créditos seleccionados: 27.00 de 23.854 créditos'. There are two input fields for 'Deuda utilizada' and 'PDA utilizada', both currently set to 0, with 'Deuda disponible: 5' and 'PDA disponible: 0' below them.
- Asignaturas:** A list of selected courses, each with a checkbox and a checkmark. The courses are:
 - [MDL] Matemática Discreta y Lógica Matemática (2019-01-29/2019-05-17), Lunes: 09:00:00-10:50:00, Miércoles: 09:00:00-10:50:00, Créditos: 6.
 - [MDL] Matemática Discreta y Lógica Matemática (2018-09-12/2018-12-20), Lunes: 11:30:00-12:20:00, Martes: 11:30:00-12:20:00, Jueves: 11:30:00-12:20:00, Viernes: 12:30:00-13:20:00, Créditos: 6.
 - [MDL] Matemática Discreta y Lógica Matemática (2019-01-23/2019-05-14), Lunes: 11:30:00-12:20:00, Martes: 11:30:00-12:20:00, Jueves: 11:30:00-12:20:00, Viernes: 12:00:00-12:50:00, Créditos: 6.
 - [MDL] Matemática Discreta y Lógica Matemática (2018-09-12/2018-12-20), Lunes: 12:00:00-12:50:00.
- Estado de la elección:** A calendar view for the period '17 - 21 de sep. de 2018'. It shows a grid of days and times. The selected courses are highlighted in the grid:

	lun. 9/17	mar. 9/18	mié. 9/19	jue. 9/20	vie. 9/21
9am	[MDL] Matemática Discreta y Lógica Matemática	[MDL] Matemática Discreta y Lógica Matemática		[MAR] Métodos Numéricos en	
10am					
11am	11:30 - [MDL] Matemática	11:30 - [MDL] Matemática		11:30 - [MDL] Matemática	
12pm					
1pm					12:30 - [MDL] Matemática
2pm					
3pm					
4pm					
5pm					
6pm					
7pm					
8pm					

Figura 51: Página de la elección de docencia

Elección Docencia Filtrar asignaturas

Elección de docencia de: David de Frutos Esc...

Información

Esta elección está confirmada

Créditos seleccionados:
27.00 de 23.854 créditos

Deuda utilizada: 0
 Deuda disponible: 5

PDA utilizada: 0
 PDA disponible: 0

Asignaturas

[MDL] Matemática Discreta y Lógica Matemática
2019-01-29/2019-05-17
Lunes: 09:00:00-10:50:00
Miércoles: 09:00:00-10:50:00
Créditos: 6

[MDL] Matemática Discreta y Lógica Matemática
2018-09-12/2018-12-20
Lunes: 11:30:00-12:20:00
Martes: 11:30:00-12:20:00
Jueves: 11:30:00-12:20:00
Viernes: 12:30:00-13:20:00
Créditos: 6

[MDL] Matemática Discreta y Lógica Matemática
2019-01-23/2019-05-14
Lunes: 11:30:00-12:20:00
Martes: 11:30:00-12:20:00
Jueves: 11:30:00-12:20:00
Viernes: 12:00:00-12:50:00
Créditos: 6

[MDL] Matemática Discreta y Lógica Matemática
2018-09-12/2018-12-20
Lunes: 12:00:00-12:50:00

Estado de la elección

Figura 52: Página elección de docencia - Estado de la elección minimizado

Elección Docencia Filtrar asignaturas

Elección de docencia de: David de Frutos Esc...

Información

Esta elección está confirmada

Créditos seleccionados:
27.00 de 23.854 créditos

Deuda utilizada: 0
 Deuda disponible: 5

PDA utilizada: 0
 PDA disponible: 0

Estado de la elección

17 – 21 de sep. de 2018 Mes | Semana | Día

	1er Cuatri.	2º Cuatri.			
	17	18	19	20	21
9am	[MDL] Matemática Discreta y Lógica Matemática	[MDL] Matemática Discreta y Lógica Matemática		[MAR] Métodos algorítmicos en programación de computadores	
10am					
11am	[MDL] Matemática Discreta y Lógica Matemática	[MDL] Matemática Discreta y Lógica Matemática		[MDL] Matemática Discreta y Lógica Matemática	
12pm					
1pm					[MDL] Matemática Discreta y Lógica Matemática
2pm					
3pm					
4pm					
5pm					
6pm					
7pm					
8pm					

Figura 53: Página elección de docencia - Asignaturas minimizadas

7.2.4.1. Listado de asignaturas y filtros

En el listado se muestra el nombre y siglas de las asignaturas, fechas entre las que se imparte, horario y créditos (se ha procurado alcanzar la mejor experiencia de usuario posible y se ha resumido la información más relevante de las asignaturas). Además, al situar el cursor por encima de cada asignatura se muestra al usuario una descripción emergente con el nombre completo de la misma.

Como se ha explicado en el apartado 7.2.3.2, existen tres tipos de asignaturas, la asignatura normal (figura 54), el desdoble (figura 55) y la divisible (figura 56).

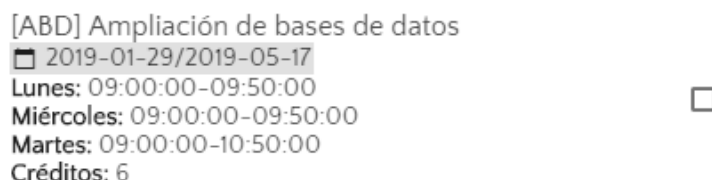


Figura 54: Elección de docencia - Tarjeta de la asignatura normal

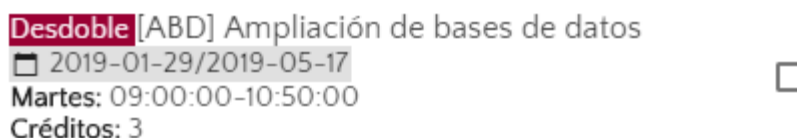


Figura 55: Elección de docencia - Tarjeta del desdoble

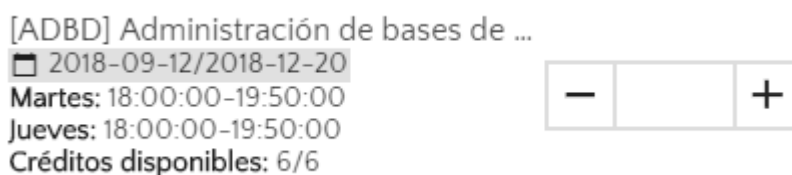


Figura 56: Elección de docencia - Tarjeta de la asignatura divisible

Según el tipo de asignatura, la forma de incluirla en la docencia varía, si se trata de una asignatura normal o un desdoble se incluye con un *checkbox*. En cambio, si se trata de una asignatura divisible, se seleccionan los créditos con un campo numérico en el que podrá pulsar sobre el símbolo “+” para sumar créditos, y el símbolo “-” para restarlos. En todo caso el máximo será el número de créditos que queden disponibles.

A continuación, se describen las acciones que se desencadenan al elegir o modificar la elección docente:

1. Cada asignatura tiene una estructura diferente, y por ello la primera función en ejecutarse dependerá del tipo de la asignatura (ver ejemplo de asignatura normal en la figura 57). Esto a su vez provoca que el número total de créditos seleccionados se actualice.

```

onSelectAsignatura(asignatura, { selected }) {
  if (this.asignaturaDisponible(asignatura)) {
    if (selected) {
      this.asignaturasSelected = [...this.asignaturasSelected, asignatura];
      this.asignaturas[this.asignaturas.indexOf(asignatura)].selected = true;
      this.creditos += asignatura.creditos;
    }
    else {
      this.asignaturasSelected = this.asignaturasSelected.filter(assign => assign.id !== asignatura.id);
      this.asignaturas[this.asignaturas.indexOf(asignatura)].selected = true;
      this.creditos -= asignatura.creditos;
    }
    this.comprobarEleccion(this.updateEleccion());
  }
}

```

Figura 57: Fragmento de código - Elección de docencia - Función ejecutada tras seleccionar una asignatura normal

2. Se actualiza la elección con los nuevos valores, de esta manera el calendario detectará el cambio y modificará las asignaturas según la nueva selección (ver figura 58). Entre los distintos componentes de Angular, la inmutabilidad de los objetos es fundamental, dado que si el objeto es el mismo no detectaría los cambios.

```

updateEleccion(saveLocal) {
  this.eleccion.asignaturas = [];
  this.eleccion.asignaturas = this.asignaturasSelected;

  this.eleccion.asignaturas_divisibles = [];
  this.eleccion.asignaturas_divisibles = this.asignaturasDivisiblesSelected;

  this.eleccion.desdobles = [];
  this.eleccion.desdobles = this.desdoblesSelected;
  this.loading = false;
  this.eleccion.deuda = this.creditosDeuda;
  saveLocal ? this.updateLocalStorage(this.eleccion) : null;
  return this.eleccion;
}

```

Figura 58: Fragmento de código - Elección de docencia - Actualizar elección

3. Se manda una petición a la API para que compruebe los posibles errores en elección (ver figura 59).

```

comprobarEleccion(eleccion) {
  const { asignaturas, desdobles, asignaturas_divisibles, deuda } = eleccion;
  this.eleccionService.comprobarEleccion(eleccion)
    .subscribe(errores => {
      this.errores = errores;
      const { L, M, X, J, V } = errores;
      this.valida = L = null
        && M = null
        && X = null
        && J = null
        && V = null;

      if (!this.valida) this.avisosService.enviarMensaje("Se han encontrado problemas en la elección")
      this.eleccion.asignaturas = asignaturas;
      this.eleccion.desdobles = desdobles;
      this.eleccion.asignaturas_divisibles = asignaturas_divisibles;
    });
}

```

Figura 59: Fragmento de código - Elección de docencia - Comprobación de la docencia

4. (Opcional) En caso de que haya algún error en la elección se le mostrará al usuario un mensaje con aquello que provoque conflictos (ver figura 60).



Figura 60: Elección de docencia - Error en la elección

El sistema de filtrado es muy similar al de los módulos anteriores, especialmente al de asignaturas. La diferencia más sustancial es que en este caso se trata de un menú que se despliega al pulsar sobre el botón “Filtrar asignaturas” (ver figura 61). En este apartado se omitirán los detalles técnicos de la búsqueda en sí porque ya han sido explicados en el apartado 7.2.2 que describe el módulo Asignaturas. Sin embargo, cabe destacar la funcionalidad de “Mostrar asignaturas disponibles”. Al pulsar sobre este interruptor se eliminan del listado las asignaturas que se encuentren incluidas en las docencias de otros profesores.

Buscar ×

Solo asignaturas disponibles
 Solo docencia asignada al profesor

Buscar por nombre

Siglas

Código

Curso

Cuatrimestre

Día(s)

Lunes
 Martes
 Miércoles
 Jueves
 Viernes

Filtrar 🔍

Figura 61: Elección de docencia - Filtros de las asignaturas

En la figura 62 se muestra el fragmento de código que comprueba si una asignatura está disponible o no. En caso de no estarlo, no se le permitirá al usuario seleccionarla y tendrá un aspecto grisáceo para diferenciarla. Si lo que desea el usuario es hacer uso del filtro de asignaturas disponibles se eliminarán de la lista aquellas que cumplan el requisito de estar disponibles para ser escogidas.

```

checkDisponibilidad(asignatura) {
  asignatura.disponible = this.asignaturaDisponible(asignatura)
  asignatura.desdobles.map(desdoble => {
    desdoble.disponible = this.asignaturaDisponible(desdoble);
  })
}

asignaturaDisponible(asignatura): boolean {
  if (asignatura.docencia.length > 0 && asignatura.docencia[0].profesor !== this.profesor.usuario.id) {
    return false;
  }
  else if (asignatura.docencia_divisible && asignatura.docencia_divisible.length > 0) {
    var userInDocencia = false
    asignatura.docencia_divisible.map(docencia => {
      userInDocencia = true ? docencia.profesor === this.profesor.usuario.id : null;
    });
    return userInDocencia
  }
  return true;
}

```

Figura 62: Fragmento de código - Elección de docencia - Asignatura disponible

7.2.4.2. Información y mensajes

El panel de información muestra al usuario el estado de su elección, es decir, si ha sido confirmada o no y el número de créditos seleccionados hasta el momento. También, se permite elegir la deuda y la PDA que se desea utilizar. En la figura 63 se puede ver un ejemplo.

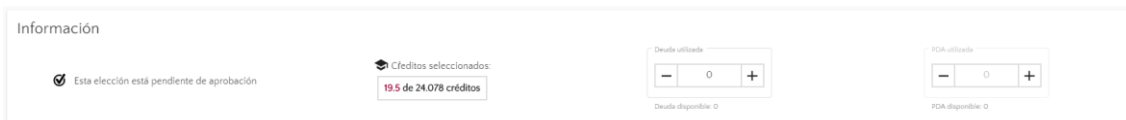


Figura 63: Elección de docencia - Panel de información

Si el usuario identificado en la aplicación tiene permisos de administración, se le mostrará encima de la información el mensaje que el usuario haya podido añadir junto a la elección en caso de que éste lo haya considerado necesario (ver figura 64).



Figura 64: Elección de docencia - Mensaje del profesor

7.2.4.3. Calendario

El componente del calendario es una parte fundamental de la aplicación dado que aporta una visión general y actualizada de las asignaturas elegidas por el profesor. Con este propósito se ha buscado una gama de colores suave para identificar cada asignatura. A cada asignatura se le asigna un color aleatorio. En las figuras 65, 66 y 67 se puede observar un ejemplo de la vista semanal, mensual y diaria de una docencia, respectivamente.

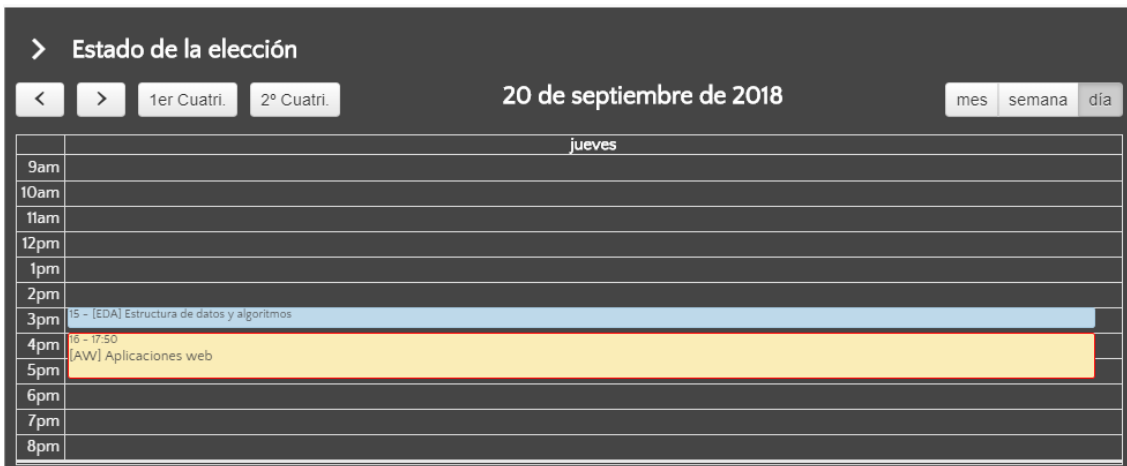


Figura 67: Elección de docencia - Vista diaria del calendario

El calendario ha sido implementado con apoyo de la librería FullCalendar [32]. Sin embargo, ha sido configurado para adaptarse a las necesidades de la aplicación, tanto el estilo como los parámetros de este (ver figura 68).

```

ngOnInit() {
  this.calendarOptions = {
    timeFormat: 'H(:mm)',
    editable: false,
    allDaySlot: false,
    locale: 'es',
    timezone: 'local',
    weekends: false,
    slotDuration: moment.duration(1, 'h'),
    minTime: moment.duration(9, 'h'),
    maxTime: moment.duration(21, 'h'),
    customButtons: {
      primerCuatrimestre: {
        text: '1er Cuatri.',
        click: () => this.moveToCuatrimestreDate(1)
      },
      segundoCuatrimestre: {
        text: '2º Cuatri.',
        click: () => this.moveToCuatrimestreDate(3)
      }
    },
    header: {
      left: 'prev, next, primerCuatrimestre, segundoCuatrimestre',
      center: 'title',
      right: 'month, agendaWeek, agendaDay'
    },
    buttonText: {
      month: 'mes',
      week: 'semana',
      day: 'día',
      list: 'lista'
    },
    events: []
  }
}

```

Figura 68: Fragmento de código - Elección de docencia - Configuración del calendario

Los tipos de datos que representan fechas han sido manipulados con la ayuda de la librería Moment.js [33]. Cada vez que se elige una asignatura, se actualiza el calendario creando un evento por cada día que se imparte la asignatura.

El calendario incluye dos botones “1er Cuatri.” y “2º Cuatri.”. Cada uno de ellos dirige la vista del calendario al cuatrimestre correspondiente. Gracias a estos atajos, el usuario puede ver fácilmente qué asignaturas tiene incluidas en su docencia a lo largo del año.

Al igual que en el filtrado de asignaturas, este componente está vinculado a un objeto del componente principal que es el de la elección de docencia. Así cada vez que haya un cambio, automáticamente Angular envía un evento, siempre que el objeto entero haya cambiado (de ahí la importancia de la inmutabilidad), que este recibe y actúa en consecuencia (ver figura 69).

```
ngOnChanges(changes: SimpleChanges) {
  const asign: SimpleChange = changes.asignaturasEntrada;
  if (asign) {
    this.asignaturas = asign.currentValue;
    if (this.asignaturas.length > 0) {
      this.updateCalendarDateView(this.asignaturas[this.asignaturas.length - 1].calendario.fecha_ini)
    }
  }
  const desd: SimpleChange = changes.desdoblesEntrada;
  if (desd) {
    this.desdobles = desd.currentValue;
    if (this.desdobles.length > 0) {
      this.updateCalendarDateView(this.desdobles[this.desdobles.length - 1].calendario.fecha_ini)
    }
  }
  const asignD: SimpleChange = changes.asignaturasDivisiblesEntrada;
  if (asignD) {
    this.asignaturasDivisibles = asignD.currentValue;
    if (this.asignaturasDivisibles.length > 0) {
      this.updateCalendarDateView(this.asignaturasDivisibles[this.asignaturasDivisibles.length - 1].asignatura.calendario.fecha_ini)
    }
  }
  this.recurEvents(this.asignaturas, this.desdobles, this.asignaturasDivisibles);
}
```

Figura 69: Fragmento de código - Elección de docencia - Función de cambio del calendario

7.2.4.4. Guardar o confirmar la docencia

A través de los botones flotantes que existen en esta vista (figura 70), el usuario accede a las funcionalidades. Cada una de ellas se realiza en la API puesto que desde la aplicación web solo se envían las peticiones.



Figura 70: Elección de docencia - Conjunto de botones de un profesor

En el caso de un profesor, existen tres posibilidades. La primera le permite restaurar la última elección guardada. La segunda le permite limpiar la elección de docencia completa pero no la guarda. En caso de que haya terminado la elección puede enviarla al administrador para que la confirme. Entonces pulsará sobre el tercer botón que desplegará una ventana emergente desde la cual podrá mandar un mensaje (ver figura 71).

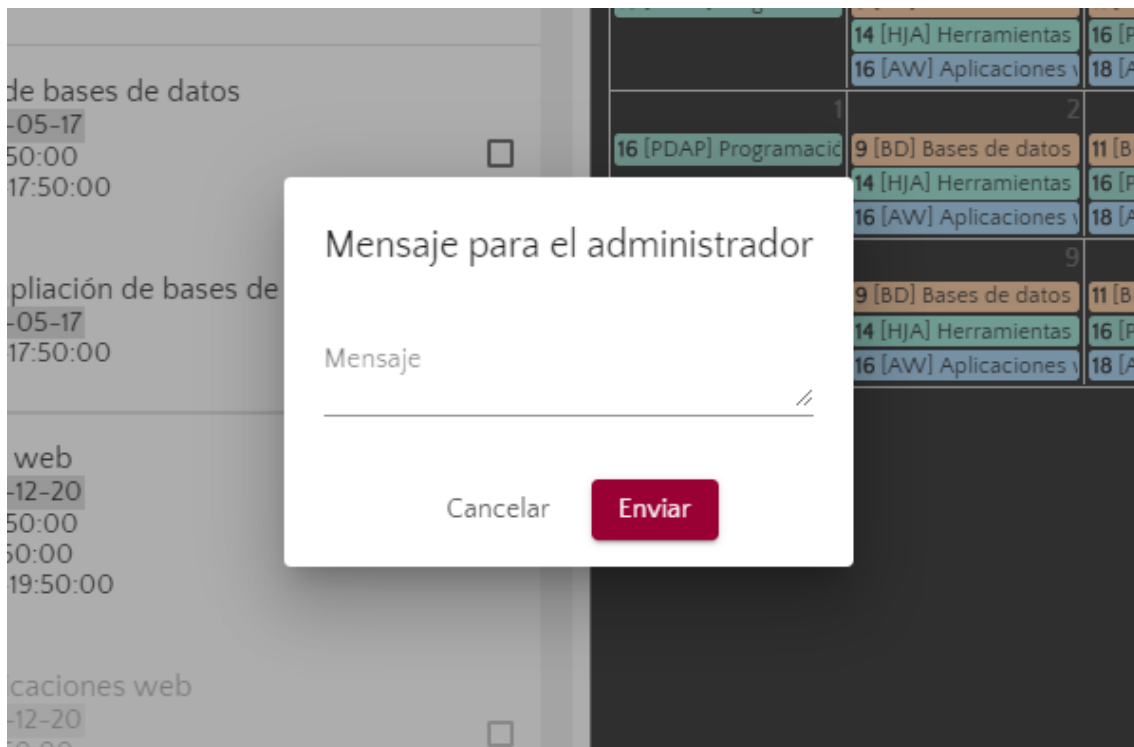


Figura 71: Elección de docencia - Mensaje para el administrador

```

saveEleccion() {
  this.updateEleccion();
  if (!this.loading) {
    if (this.admin) {
      this.eleccion.confirmada = true;
    }
    if (this.eleccion.id !== undefined) {
      this.eleccionService.saveEleccion(this.eleccion);
    }
    else {
      this.eleccionService.createEleccion(this.eleccion).subscribe(data => {
        this.avisosService.enviarMensaje("Elección de docencia guardada correctamente");
        window.location.reload()
      });
    }
  }
}

```

Figura 72: Fragmento de código - Elección de docencia - Guardar o crear elección

Como se puede ver en la figura 72, en caso de que la elección no esté ya creada se pide a la API que la cree. En caso contrario se le pedirá que la guarde.

Observar que, si el usuario identificado tuviera permisos de administración, podrá ver cuatro botones en la parte inferior derecha (ver figura 73).



Figura 73: Elección de docencia - Conjunto de botones de un administrador

El primero de ellos elimina completamente la elección del profesor. La funcionalidad se desarrolla en la API. Así desde el *frontend* solo se envía una petición, y una vez finalizada se le muestra un mensaje al usuario (ver figura 74).

```
deleteEleccion() {
  this.loading = true;
  this.deleteLocalStorage(this.profesor.usuario.id)
  this.eleccionService.deleteEleccion(this.eleccion.id).subscribe(
    () => {
      this.clearEleccion();
      this.reset();
      this.updateProfesores();
      this.avisosService.enviarMensaje("Elección de docencia eliminada correctamente");
    },
    err => {
      this.avisosService.enviarMensaje(
        "Error al eliminar la docencia, prueba a volver a la última versión guardada, es posible que haya sido editada");
      this.loading = false;
      this.updateLocalStorage(this.eleccion)
    },
    () => {
      this.loading = false;
    }
  );
}
```

Figura 74: Fragmento de código - Elección de docencia - Eliminar elección

El segundo restaura la elección docente a la última que se haya guardado.

El tercero, al igual que con el profesor normal, limpia toda la elección pendiente de ser guardada.

El cuarto botón es el encargado de guardar la docencia. En este sentido, guarda los cambios realizados en la elección de docencia como ocurre en el caso del usuario, salvo que en esta ocasión el administrador puede decidir si guardar la docencia confirmada o sin confirmar. Tras la confirmación de la docencia el turno cambiará al siguiente profesor por orden de escalafón.

7.2.5. Autenticación

En esta sección se va a tratar el tema de la autenticación en la aplicación. La autenticación de los usuarios se realiza en el lado del servidor, de manera que se describirá la pantalla de *login*, la gestión del token de autenticación y las restricciones que existen al navegar entre las páginas según los diferentes tipos de usuario.

7.2.5.1. Login y gestión del token

El sistema de autenticación es uno de los pilares de la aplicación, dado que éste depende la identificación de los usuarios. En la figura 75 se muestra una captura de la pantalla del *login*. Como se puede observar, se ha buscado continuar con el mismo estilo de diseño que en el resto de la aplicación.

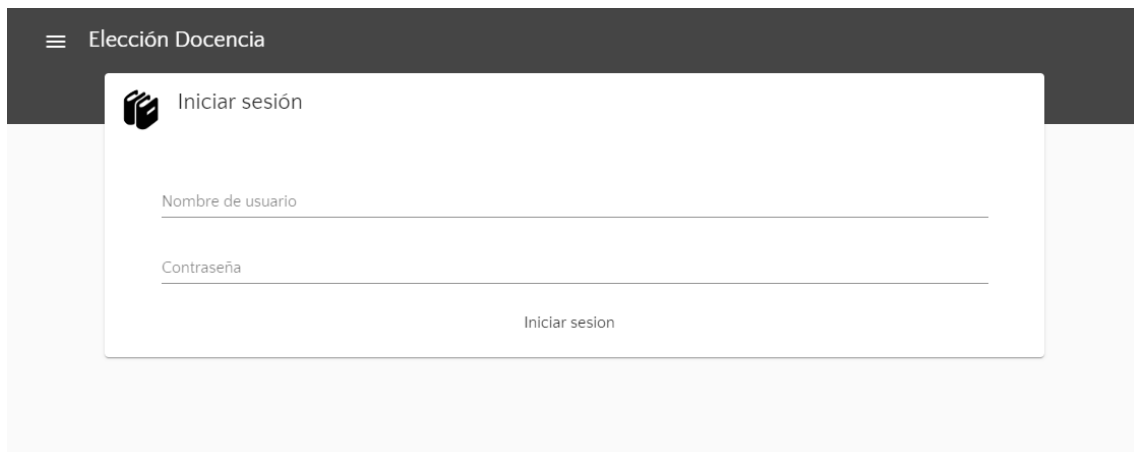


Figura 75: Elección de docencia - Pantalla de login

Una vez el usuario introduce sus credenciales, se realiza una petición a la API con los datos. En caso de que sean correctos, devolverá dos tokens, uno que identifica al usuario *logueado*, y otro que se usará para refrescar la sesión si esta caducase. En la figura 76 se puede observar el fragmento de código relacionado con la funcionalidad descrita. Ambos se guardarán en el almacenamiento local del navegador.

```
login(usuario, password) {  
  let credentials = {  
    username: usuario,  
    password: password  
  }  
  
  return this.http.post<Token>("/api/token/", credentials);  
}
```

Figura 76: Fragmento de código - Elección de docencia - Login

A partir de este momento entra en juego el interceptor de peticiones y errores. Cada vez que la aplicación solicita algo a la API, este servicio lo intercepta, y le añade la URL de la API y el token de la sesión (ver figura 77).

```
private addTokenToRequest(request: HttpRequest<any>, token: string) : HttpRequest<any> {  
  const url = 'http://docenciafdi.me';  
  return request.clone({ url: url + request.url, setHeaders: { Authorization: `Bearer ${token}`}});  
}
```

Figura 77: Fragmento de código - Elección de docencia - Añadir token a las peticiones

Si la petición enviada a la API devuelve un error, entonces el interceptor lo captura y lo trata según el tipo. En la figura 78 se puede observar un fragmento de la función que se encarga de controlar estos errores y mostrarlos por pantalla en caso de que se requiera.

```

switch ((<HttpErrorResponse>err).status) {
  case 401:
    return this.handle401Error(request, next);
  case 403:
    this.avisosService.enviarMensaje("No tienes permisos para realizar la acción");
    return empty();
  case 0:
    this.avisosService.enviarMensaje("Hay problemas de conexión con el servidor");
    return empty();
  default:
    return throwError(err);
}

```

Figura 78: Fragmento de código - Elección de docencia - Fragmento del filtrado de errores

Si el error es del tipo 401 Error de autenticación [34], entonces se comprobará si existe un token para refrescar la sesión. En caso afirmativo se ejecutará el refresco, y en caso contrario no se le permitirá acceder al usuario.

7.2.5.2. Cerrar sesión

La funcionalidad de cerrar sesión (figura 79) no requiere comunicación directa con la API. Para hacerla efectiva se borra el almacenamiento local donde se guardan los dos tipos de token y se le redirige al usuario a la pantalla principal de la aplicación. No es necesario ningún tipo de comunicación con el servidor, ya que los tokens tienen una fecha de caducidad relativamente corta.

Como todas las pantallas requieren autenticación (tal como se explica en el siguiente apartado) entonces desde la pantalla principal se le redirige a la pantalla de *login*.

```

logout() {
  // remove user from local storage to log user out
  localStorage.clear()
  this.router.navigate(['/dashboard/'])
  this.name.next("Iniciar sesión");
  return empty();
}

```

Figura 79: Fragmento de código - Elección de docencia - Logout

7.2.5.3. Restricciones en la navegación

Como ya se ha introducido en el apartado anterior, todas las páginas, salvo la de inicio de sesión, necesitan de un usuario identificado para ser accesibles. En la figura 80 se puede ver el archivo de configuración que logra *securizar* las distintas rutas de la aplicación.

```

const routes: Routes = [
  {
    path: '',
    children: [
      { path: '', loadChildren: './auth/auth.module#AuthModule' },
      { path: 'dashboard', component: DashboardComponent, canActivate: [AuthGuardService] },
      { path: 'config', component: ConfigurationComponent, canActivate: [AuthGuardService] },
      { path: 'mi-cuenta', component: ProfesoresDetailsComponent, canActivate: [AuthGuardService] },
    ]
  }
];

```

Figura 80: Fragmento de código - Elección de docencia - Ejemplo de restricción de las rutas

De esta manera, cada vez que un usuario accede a una ruta, se comprobará que tiene los permisos necesarios para acceder a ella.

Hay una serie de rutas que tienen una restricción especial, solo son accesibles por los administradores. En este sentido, cada vez que se realiza la comprobación descrita anteriormente, se añade otra comprobación que consiste en acceder al archivo de rutas restringidas y comparar la ruta actual con las que contiene este. (Figura 81).

```

1  export default [
2      "add-asignatura/:id",
3      "edit-asignatura/:id",
4      "importar-asignaturas",
5      'add-profesor/:id',
6      'edit-profesor/:id',
7      'importar-profesores',
8      'config'
9  ]

```

Figura 81: Fragmento de código - Elección de docencia - Rutas restringidas

7.3 API REST

En este apartado se especificará el diseño de la API REST implementada. Como en el apartado anterior, las funcionalidades serán agrupadas por módulos.

La decisión de escoger Django como *framework* para el desarrollo de la API se ha tomado principalmente por la rapidez en el desarrollo, la facilidad de *securizar* la aplicación antes los ataques más comunes (CSRF, *SQL Injection*, XSS, *Clickjacking*, etc.), la posibilidad de escalar la aplicación de manera sencilla, y la extensa documentación que proporciona.

En las primeras fases del desarrollo se ha utilizado el servidor web incluido en Django. Sin embargo, debido a las limitaciones que presenta el servidor de Django, se utilizó para alojar la API un servidor Apache con el módulo *mod_wsgi* que permite comunicarse con Django a través de la Web Server Gateway Interface [35] (WSGI)¹.

¹ WSGI es una interface simple y universal entre los servidores web y las aplicaciones web o frameworks.

Además, se ha creado un entorno virtual en Python con *virtualenv* con la intención de mantener un control de las dependencias necesarias del proyecto como de la versión del intérprete.

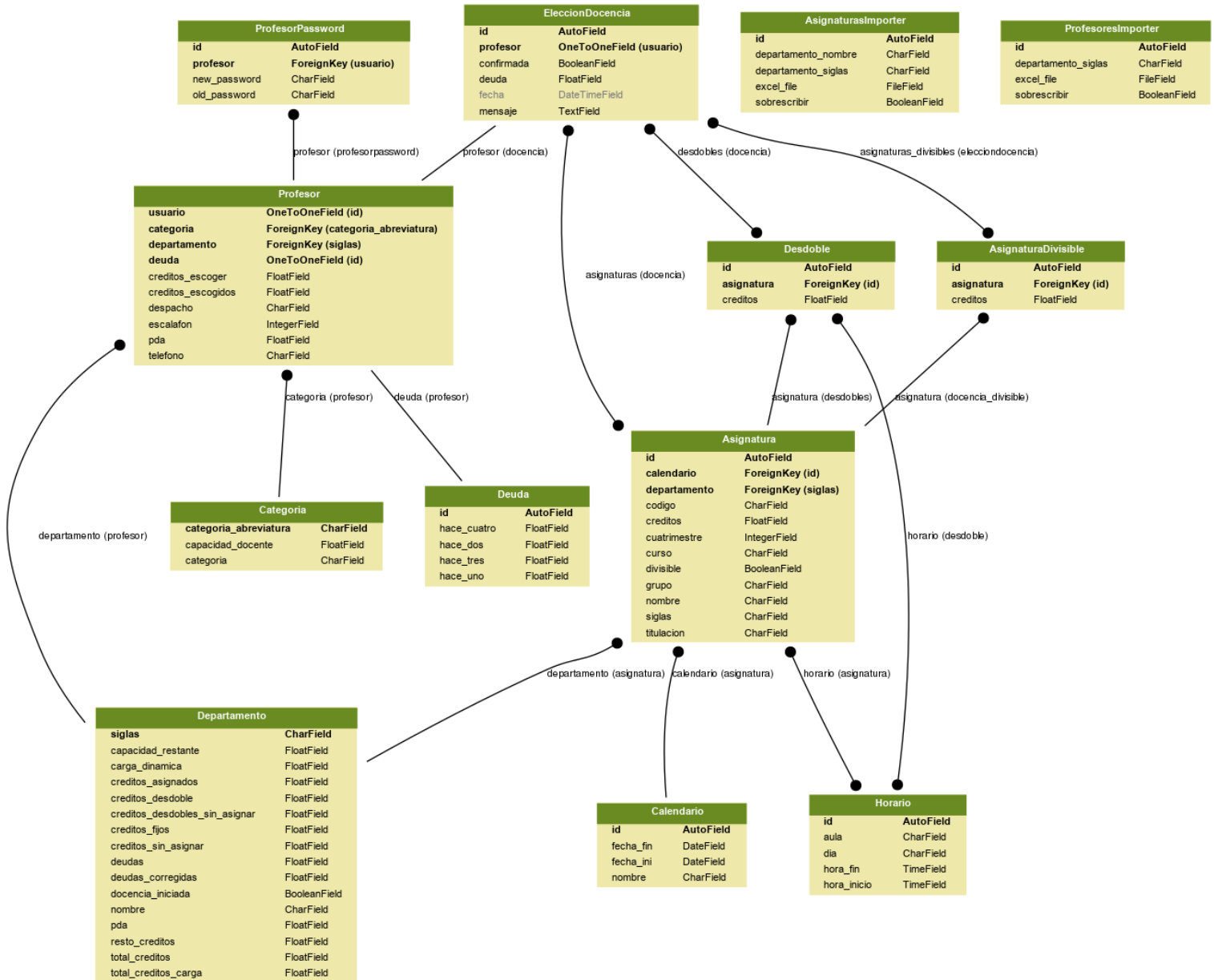


Figura 82: Diagrama de clases de la API

El diseño de clases de la API es el que se muestra en la figura 82. Como se puede apreciar es prácticamente igual al diagrama de base de datos que se muestra en las figuras del capítulo ‘Modelo de datos’. Django genera automáticamente el esquema de la base de datos a partir de las clases que se definan en sus aplicaciones².

² Un proyecto de Django está formado de un conjunto de aplicaciones que se comunican entre sí dando lugar al sistema completo.

En el sistema implementado existen cuatro aplicaciones distintas. La aplicación de asignaturas que define las clases: *Asignatura*, *Desdoble*, *Calendario*, *Horario* y *AsignaturaImporter*. A continuación, la aplicación de profesores que define las clases: *Profesor*, *Categoria*, *Deuda*, *ProfesoresImporter* y *ProfesorPassword*. Después, la aplicación de Elección de docencia que define las clases: *EleccionDocencia* y *AsignaturaDivisible*. Y, por último, la aplicación de Departamentos que solamente contiene a la clase *Departamento*.

El sistema consta de más aplicaciones que han sido incorporadas como las relacionadas con la autenticación [36], sesiones [37], control de acceso HTTP (CORS) [38], o herramientas de filtrado de consultas [39].

A continuación, se describirán exclusivamente las aplicaciones relativas al sistema de elección de docencia implementado.

7.3.1. Asignaturas

En la tabla 1 que se muestra a continuación se observa una relación de los *endpoints* y operaciones CRUD que se pueden realizar a través de los métodos HTTP.

Endpoint	GET	POST	PUT	PATCH	DELETE
/api/asignaturas/	X	X			
/api/asignaturas/importar/		X			
/api/asignaturas/{id}/	X		X	X	X
/api/asignaturas/calendarios/	X	X			
/api/asignaturas/calendarios/{id}/	X		X	X	X
/api/asignaturas/{asignatura_pk}/desdobles/	X	X			
/api/asignaturas/{asignatura_pk}/desdobles/{id}/	X		X	X	X
/api/asignaturas/{asignatura_pk}/desdobles/{desdoble_pk}/horarios/	X	X			
/api/asignaturas/{asignatura_pk}/desdobles/{desdoble_pk}/horarios/{id}/	X		X	X	X
/api/asignaturas/exportar/	X				
/api/asignaturas/{asignatura_pk}/horarios/	X	X			
/api/asignaturas/{asignatura_pk}/horarios/{id}/	X		X	X	X

Tabla 1: Endpoints API - Asignaturas

Parámetros de consulta para el filtrado de asignaturas

Los parámetros que se muestran en la tabla 2 deben ser incluidos como parte de la URL.

Parámetro	Descripción
<i>limit</i>	Número de resultados que se muestran en cada respuesta
<i>offset</i>	Índice inicial desde donde se muestran resultados
<i>siglas</i>	Siglas de la asignatura
<i>nombre</i>	Nombre de la asignatura
<i>codigo</i>	Código GEA
<i>curso</i>	Curso de la asignatura
<i>cuatrimestre</i>	Cuatrimestre en el que se imparte
<i>dia</i>	Excluye asignaturas que se impartan en el día de la semana <i>dia</i>
<i>inicio</i>	Excluye las asignaturas las cuales en alguno de sus horarios su hora de inicio sea <i>inicio</i>
<i>fin</i>	Excluye las asignaturas las cuales en alguno de sus horarios su hora de finalización sea <i>fin</i>
<i>desdobles</i>	Identificadores de desdobles
<i>docencia</i>	Booleano para filtrar por el estado de la docencia de la asignatura (asignada o no)
<i>profesor</i>	Identificador de profesor para filtrar por las asignaturas que tiene un profesor incluidas en su docencia
<i>profesor_nombre</i>	Nombre del profesor
<i>profesor_apellido</i>	Apellido del profesor

Tabla 2: Parámetros de consulta API - Asignaturas

Dependiendo del método HTTP que se utilice en cada *endpoint* se invoca una operación u otra. El método GET si se utiliza sobre un recurso individual se realiza una operación de lectura (*read*), mientras que se si utiliza sobre un conjunto de recursos se realiza una operación de listado (*list*). El método POST corresponde con la operación de creación de recursos y solamente se puede invocar sobre un listado de recursos. El método

PUT (*update*) y PATCH (*partial update*) son muy similares, el primero sustituye un recurso por otro, y el segundo permite modificar algunos de los campos de un recurso. Por último, el método DELETE se puede utilizar sobre recursos individuales y permite eliminarlos.

A continuación, se muestra un resumen de la respuesta de la API tras realizar una petición GET:

```
GET /api/asignaturas/
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[
  {
    "id": 0,
    "departamento": "SIC",
    "horario": [
      {
        "id": 0,
        "dia": "M",
        "hora_inicio": "18:00:00",
        "hora_fin": "19:50:00",
        "aula": "Aula 7"
      },
      {
        "id": 1,
        "dia": "J",
        "hora_inicio": "18:00:00",
        "hora_fin": "19:50:00",
        "aula": "Aula 7"
      }
    ],
    "desdobles": [],
    "calendario": {
      "id": 1,
      "nombre": "Grados y máster",
      "fecha_ini": "2018-09-12",
      "fecha_fin": "2018-12-20"
    }
  }
]
```

```
    },  
    "docencia_divisible": [],  
    "siglas": "AEDB",  
    "nombre": "Administración de bases de datos",  
    "grupo": "A",  
    "cuatrimestre": 1,  
    "creditos": 6.0,  
    "codigo": "607300",  
    "curso": "2º",  
    "titulacion": "MII",  
    "divisible": true,  
    "docencia": []  
  }, ...
```

En la figura 83 se muestra un fragmento de un *serializer*. Los *serializers* se encargan de transformar estructuras de datos complejas e instancias de clases en tipos nativos de Python que son fácilmente representados en formatos JSON o XML. También son capaces de hacer la transformación en el sentido inverso, después de validar los datos entrantes.

```
class AsignaturaSerializer(serializers.ModelSerializer):
    q_departamento = Departamento.objects.all()
    departamento = serializers.PrimaryKeyRelatedField(
        queryset=q_departamento, many=False, read_only=False)

    horario = HorarioSerializer(many=True)
    desdobles = DesdobleSerializer(many=True)
    calendario = CalendarioSerializer()

    # Serializer que muestre campo con el profesor de la docencia divisible
    docencia_divisible = serializers.SerializerMethodField()

    def get_docencia_divisible(self, obj):
        from eleccion_docencia.serializers import AsignaturaDivisibleProfesorSerializer
        from eleccion_docencia.models import AsignaturaDivisible

        qs = AsignaturaDivisible.objects.filter(asignatura=obj)

        return AsignaturaDivisibleProfesorSerializer(qs, many=True, read_only=True).data

You, 3 months ago | 1 author (You)
class Meta:
    model = Asignatura
    fields = '__all__'
    extra_fields = ['desdobles', 'docencia', 'docencia_divisible']
    depth = 1

def create(self, validated_data):
    horarios_data = validated_data.pop('horario')
    desdobles_data = validated_data.pop('desdobles')
    calendario_data = validated_data.pop('calendario')

    if 'id' in calendario_data:
        calendario = Calendario.objects.get(id=calendario_data['id'])
        # No actualizamos los demás campos si se encuentra el id
        # Un cambio de un calendario concreto en una asignatura se vería reflejado en todos los demás
    else:
        calendario = Calendario.objects.create(**calendario_data)

    asignatura = Asignatura.objects.create(
        calendario=calendario, **validated_data)

    for horario_data in horarios_data:
        horario = Horario.objects.create(
            asignatura=asignatura, **horario_data)
        asignatura.horario.add(horario)

    for desdoble_data in desdobles_data:
        desdoble = Desdoble.objects.create(
            asignatura=asignatura, creditos=desdoble_data['creditos'])
        for horario in desdoble_data['horario']:
            horario = Horario.objects.create(**horario)
            desdoble.horario.add(horario)
            # ¿Añadir desdoble a la asignatura?

    return asignatura
```

Figura 83: Fragmento de código - Asignaturas serializer

Los *viewsets* permiten agrupar la lógica de un conjunto vistas relacionadas entre sí en una sola clase. En otros *frameworks* se corresponde a lo que se conoce como controladores. Los métodos que proporciona un *viewset* son *list()*, *create()*, *retrieve()*, *update()*, *partial_update()* y *destroy()*, y están directamente relacionados con los métodos HTTP disponibles. En la figura 84 se puede ver un fragmento del *viewset* de las asignaturas.

```
class AsignaturaViewSet(viewsets.ModelViewSet):
    queryset = Asignatura.objects.all()
    serializer_class = AsignaturaSerializer
    permission_classes = (IsAuthenticated, AsignaturasPermissions)
    filter_backends = (DjangoFilterBackend,)
    filterset_class = AsignaturaFilter
    lookup_value_regex = '[0-9]*'

    def get_serializer_class(self):
        if self.action == 'importar':
            return AsignaturaImporterSerializer
        return AsignaturaSerializer

    @action(detail=False, methods=['post'])
    def importar(self, request, format=None):
        if 'excel_file' in request.FILES and 'departamento_siglas' in request.data \
            and 'departamento_nombre' in request.data:

            serializer = AsignaturaImporterSerializer(data=request.data)

            if serializer.is_valid():

                file = serializer.save()
                file.save()
                file.importar()

                return Response(serializer.data)
            else:
                return Response(serializer.errors)
        else:
            raise ValidationError("Se deben mandar todos los campos")
```

Figura 84: Fragmento de código - Asignaturas viewset

En cuanto a la importación de asignaturas, el sistema es capaz de *parsear* una hoja de cálculo con el listado de asignaturas y poblar la base de datos con ellas. El formato de la hoja de cálculo que se importe al sistema tiene que ser estrictamente igual al que se indica en la aplicación para que funcione correctamente. Sin embargo, se ha procurado que éste sea muy similar al que se utiliza en los departamentos de la facultad. A continuación, se muestra un ejemplo de una petición al recurso que importa asignaturas al sistema:

```
POST /api/asignaturas/importar/

HTTP 200 OK

Allow: POST, OPTIONS

Content-Type: application/json

Vary: Accept

{
  "excel_file": "/uploads/asignaturas/horarios_fdi_amMNTTx.xlsx",
  "departamento_siglas": "SIC",
  "departamento_nombre": "Sistemas Informáticos y Computación",
  "sobrescribir": true
}
```

Es necesario informar a la aplicación acerca del departamento al que pertenecen las asignaturas que se quieren importar, ya que las hojas de cálculo suelen incluir asignaturas de otros departamentos distintos al que se quiere gestionar. Además, la primera vez que se importan asignaturas, en la aplicación también se registra el departamento al que pertenecen si éste todavía no se había creado.

En la figura 85 se muestra un ejemplo de hoja de cálculo utilizada para la importación de asignaturas.

	A	B	C	D	E	F	G
1	Tit.	Curso	Cod.GEA	Sigla	Asignatura	Grupo	Dpto.
2	MII	2º	607300	ADBD	Administración de bases de datos	A	SIC
3	GS	3º	803345	ASR	Administración de sistemas y redes	E	ACYA
4	GI	3º	803284	ABD	Ampliación de bases de datos	B	SIC
5	GI	3º	803284	ABD	Ampliación de bases de datos	C	SIC
6	GS	3º	803335	ABD	Ampliación de bases de datos	E	ISIA
7	GC	3º	803215	AR	Ampliación de Redes	D	ACYA
8	GI	4º	803290	ASOR	Ampliación de Sistemas Operativos y Redes	B	ACYA
9	GI	4º	803290	ASOR	Ampliación de Sistemas Operativos y Redes	C	ACYA
10	GI	4º	803290	ASOR	Ampliación de Sistemas Operativos y Redes	I	ACYA
11	GI/DG	4º/5º	803290/900266	ASOR	Ampliación de Sistemas Operativos y Redes	A	ACYA
12	GC	3º	803214	ASO	Ampliación de Sistemas Operativos	D	ACYA
13	GI/GC/GS	Opt.	803364	SOC	Análisis de redes sociales	A	ISIA
14	GS	4º	803337	AW	Aplicaciones web	E	SIC
15	GI/GC	3º	803285/803223	AW/SW	Aplicaciones web / Sistemas web	B	ISIA
16	GI/GC	3º	803285/803223	AW/SW	Aplicaciones web / Sistemas web	C	ISIA
17	GI/GC	3º	803285/803223	AW/SW	Aplicaciones web / Sistemas web	D	ISIA
18	GI/GC/GS/GDV/DG	Opt.	803300/900243/805318	AA	Aprendizaje automático y Big Data / Aprendizaje automático y minería de datos	A	ISIA
19	GI	4º	803289	AC	Arquitectura de Computadores	I	ACYA
20	GI/GC	4º / 3º	803218/803289	AC	Arquitectura de Computadores	D	ACYA
21	GI/GC	4º	803218/803289	AC	Arquitectura de Computadores	B	ACYA
22	GI/GC	4º	803218/803289	AC	Arquitectura de Computadores	C	ACYA
23	GI/DG/GC	4º/5º	803218/803289/900267	AC	Arquitectura de Computadores	A	ACYA

Figura 85: Hoja de cálculo de las asignaturas

```

"""Almacena en 'asignaturas' las asignaturas que haya en el libro de Excel 'worksheet'
que sean del departamento indicado por parámetro. A su vez, va conformando la lista de
horarios a medida que se recorre la lista de asignaturas.
"""

def parse_horarios(worksheet, asignaturas, departamento_find, horarios):
    first_row_passed = False # en la primera fila solo están los títulos de las columnas

    for asignatura in worksheet:
        if first_row_passed:
            if asignatura[DEPARTAMENTO_HORARIO].value.find(departamento_find) != -1:

                nombre = asignatura[ASIGNATURA_HORARIO].value
                grupo = asignatura[GRUPO_HORARIO].value
                departamento = departamento_find
                cuatrimestre = asignatura[CUATRIMESTRE_HORARIO].value
                aula = asignatura[AULA_HORARIO].value
                horario, credits = create_horario(
                    horarios, asignatura[HORARIO_HORARIO].value.strip(), aula)
                siglas = asignatura[SIGLA_HORARIO].value
                codigo = asignatura[CODIGO_HORARIO].value
                curso = asignatura[CURSO_HORARIO].value
                titulacion = asignatura[TIULACION_HORARIO].value
                divisible = True if curso in DIVISIBLES or titulacion in DIVISIBLES else False

                if cuatrimestre == 1:
                    if titulacion == "DG":
                        calendario = CALENDARIO_I_DG
                    else:
                        calendario = CALENDARIO_I_RESTO
                else:
                    if titulacion == "DG":
                        calendario = CALENDARIO_II_DG
                    else:
                        calendario = CALENDARIO_II_RESTO

                row = Asignaturas(nombre=nombre,
                                grupo=grupo,
                                departamento=departamento,
                                cuatrimestre=cuatrimestre,
                                horario=horario,

```

Figura 86: Fragmento de código - Importar asignaturas

La figura 86 ilustra un fragmento del código encargado de extraer las asignaturas de la hoja de cálculo. Primero, almacena las asignaturas, horarios y desdobles en listas de objetos, y después crea un documento JSON que en la terminología de Django se conoce como *fixture* [40]. Los *fixtures* son colecciones de datos escritas en JSON, XML o YAML y que Django es capaz de importar a la base de datos.

En la figura 87 se muestra un fragmento de un *fixture* generado durante la importación de asignaturas.

```

{
  "model": "asignaturas.asignatura",
  "pk": 27,
  "fields": {
    "nombre": "Fundamentos de la Programaci\u00f3n",
    "grupo": "A",
    "cuatrimestre": 1,
    "horario": [
      69,
      70,
      71
    ],
    "creditos": 6.0,
    "codigo": 803264,
    "curso": "1\u00ba",
    "titulacion": "GI/GC/GS",
    "siglas": "FP",
    "departamento": "SIC",
    "calendario": 1,
    "divisible": false
  }
},

```

Figura 87: Ejemplo de fixture para la importación de asignaturas

7.3.2. Profesores

Los *endpoints* disponibles en la aplicación de profesores son los que se muestran en la tabla 3:

Endpoint	GET	POST	PUT	PATCH	DELETE
/api/profesores/	X	X			
/api/profesores/importar/		X			
/api/profesores/password/		X			
/api/profesores/{usuario_id}/	X		X	X	X
/api/profesores/categorias/	X	X			
/api/profesores/categorias/{categoria_abreviatura}/	X		X	X	X
/api/profesores/exportar/	X				

Tabla 3: Endpoints API - Profesores

Parámetros de consulta para el filtrado de profesores

Los parámetros que se muestran en la tabla 4 deben ser incluidos como parte de la URL.

Parámetro	Descripción
<code>limit</code>	Número de resultados que se muestran en cada respuesta
<code>offset</code>	Índice inicial desde donde se muestran resultados
<code>usuario</code>	Nombre de usuario del profesor
<code>nombre</code>	Nombre del profesor
<code>apellido</code>	Apellido del profesor
<code>escalafon</code>	Posición en el escalafón
<code>despacho</code>	Despacho del profesor
<code>telefono</code>	Teléfono del profesor
<code>categoria</code>	Categoría del profesor
<code>docencia</code>	Booleano que indica si el profesor ya ha iniciado su elección de docencia

Tabla 4: Parámetros de consulta API - Profesores

El recurso profesor contiene una referencia al modelo de usuario que proporciona Django. De esta manera, se aprovecha el sistema de autenticación que proporciona el *framework*, y a la hora de añadir un profesor al sistema, se crea un usuario simultáneamente. Como se puede observar en la respuesta que se obtiene al solicitar la lista de profesores, el campo usuario contiene algunos de los campos que vienen definidos por el modelo de usuario de Django.

```
GET /api/profesores/
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[[
  {
    "usuario": {
      "id": 66,
      "username": "jpicator",
      "first_name": "Javier",
```

```

        "last_name": "Picatoste",
        "email": "jpicatos@ucm.es",
        "is_staff": true
    },
    "deuda": {
        "id": 64,
        "hace_uno": 0.0,
        "hace_dos": 0.0,
        "hace_tres": 0.0,
        "hace_cuatro": 0.0
    },
    "docencia": null,
    "docencia_confirmada": false,
    "escalafon": 99,
    "telefono": "666666666",
    "despacho": "0",
    "pda": 0.0,
    "creditos_escoger": 0.0,
    "creditos_escogidos": 0.0,
    "categoria": "AY",
    "departamento": "SIC"
}, ...

```

En el momento de la creación de un profesor, la API espera que el cliente mande las credenciales que le permitirán identificarse en el sistema. La decisión que se ha tomado ha sido que la aplicación web genere una contraseña temporal en el cliente. Solamente el cliente deberá generar la contraseña temporal en el momento de crear un único profesor. Cuando se importan profesores desde un fichero, es el servidor quien genera la contraseña. Posteriormente, esta contraseña temporal le será enviada al profesor por correo electrónico (ver figura 88 y 89). Se ha tomado esta decisión porque a la hora de importar profesores al sistema, no sería factible para el administrador asignar manualmente una contraseña a cada profesor.

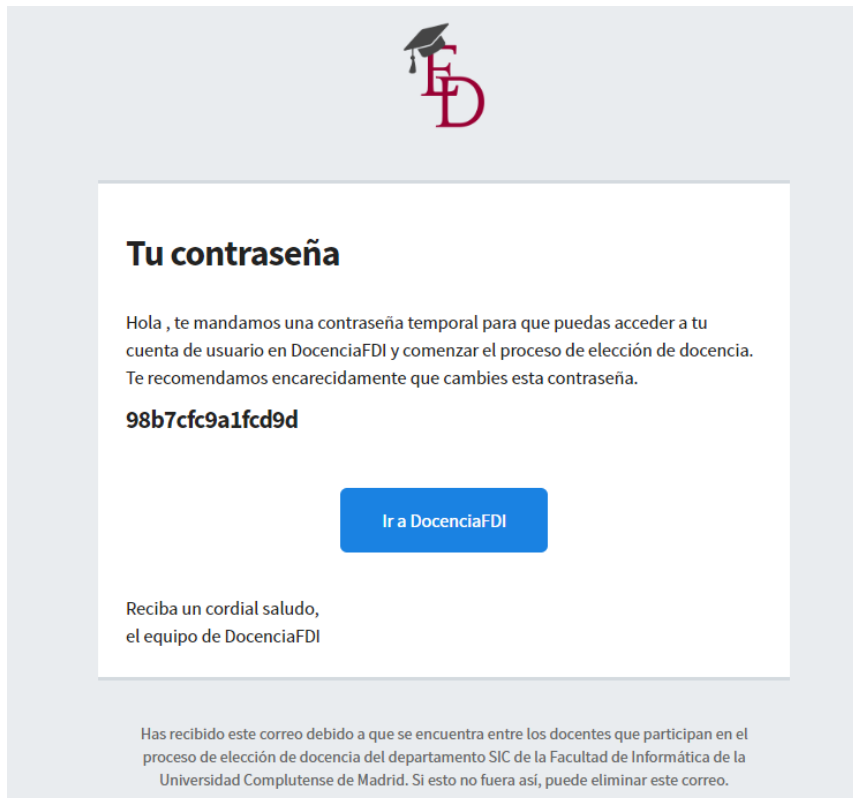


Figura 88: Correo electrónico confirmación registro con contraseña temporal

```

@staticmethod
def send_password(destination, profesor, password, departamento):
    subject = 'DocenciaFDI - Contraseña nuevo usuario'
    html_message = render_to_string('password_mail.html', {
        'profesor': profesor, 'password': password, 'departamento': departamento})
    plain_message = strip_tags(html_message)
    from_email = 'DocenciaFDI <noreply@docenciafdi.me>'
    to = destination

    mail.send_mail(subject, plain_message, from_email,
        [to], html_message=html_message)

```

Figura 89: Fragmento de código - Enviar contraseña temporal

Cuando el profesor recibe el correo, entonces debería cambiar la contraseña temporal. Para ello existe un *endpoint* específico.

```

POST /api/profesores/password/
HTTP 200 OK
Allow: POST, OPTIONS
Content-Type: application/json
Vary: Accept

```

```
{
  "profesor": 66,
  "old_password": "98b7cfc9a1fcd9d",
  "new_password": "very_strong_password"
}
```

Si se quisiera dar permisos de administración a un profesor bastaría con realizar una petición PATCH como la que se muestra a continuación:

```
PATCH /api/profesores/66/
HTTP 200 OK
Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept
{
  "usuario": {
    "id": 66,
    "is_staff": true
  }
}
```

Cuando un profesor tiene permisos de administración tiene básicamente los mismos que un *superusuario*. De esta manera podría sustituirlo cuando fuera necesario y poder llevar el control del proceso de elección de docencia, editar asignaturas y profesores.

Finalmente, cabe destacar que durante la creación de profesores se van actualizando las deudas, PDA, capacidad docente y créditos agregados del departamento al que pertenecen (ver figura 90).

```

"""
PDA
"""

pda = Profesor.objects.aggregate(Sum('pda'))['pda__sum']
departamento.pda = pda if pda else 0

"""
CRÉDITOS FIJOS (ASOCIADOS, BECARIOS Y AYUDANTES)
"""

asociados = Profesor.objects.filter(categoria__categoria__icontains='asociado').aggregate(
    Sum('categoria__capacidad_docente'))['categoria__capacidad_docente__sum']
asociados = asociados if asociados else 0
becarios = Profesor.objects.filter(categoria__categoria__icontains='becario').aggregate(
    Sum('categoria__capacidad_docente'))['categoria__capacidad_docente__sum']
becarios = becarios if becarios else 0
ayudantes = Profesor.objects.filter(categoria__categoria__icontains='ayudante').aggregate(
    Sum('categoria__capacidad_docente'))['categoria__capacidad_docente__sum']
ayudantes = ayudantes if ayudantes else 0

departamento.creditos_fijos = asociados + becarios + ayudantes

departamento.calcular() # Resto de créditos (sin fijos) y total

departamento.save()

```

Figura 90: Fragmento de código - Cálculo de PDA y créditos fijos del departamento

En cuanto a la importación de profesores, no hay muchas diferencias con respecto a la importación de asignaturas. Como también ocurre con las asignaturas, es necesario indicar el departamento al que pertenecen los profesores, y también se permite sobrescribir el listado de profesores actual o añadir nuevos profesores al sistema. En la figura 91 se muestra un ejemplo de hoja de cálculo utilizada para la importación de profesores.

Como se ha indicado antes, la generación de contraseñas en el momento de la importación se realiza en el lado del servidor y no en el cliente como ocurre cuando se registra un nuevo profesor desde el formulario que existe en la aplicación web.

	A	B	C	D	E	F
1	escalafón	apellidos, nombre	cat.	teléfono	despacho	e-mail
2	1	Frutos Escrig, David de	CU	4529	310C(Mat)	defrutos@sip.ucm.es
3	2	Peña Marí, Ricardo	CU	7627	418	ricardo@sip.ucm.es
4	3	Martí Oñiet, Narciso	CU	7557	441	narciso@sip.ucm.es
5	4	López Fraguas, Francisco J.	CU	7630	423, 406(Dirección)	fraguas@sip.ucm.es
6	5	Núñez García, Manuel	CU	7628	419	mn@sip.ucm.es
7	6	Fernández Camacho, María Inés	TU	4238	310B(Mat)	minesfc@sip.ucm.es
8	7	Bradley Delso, Margarita	TU	4264	310D(Mat)	bradley@sip.ucm.es
9	8	Gil Luezas, Ana	TU	7632	425	anagil@sip.ucm.es
10	9	Ortega Mallén, Yolanda	TU	4467	310E(Mat)	yolanda@sip.ucm.es
11	10	Gavilanes Franco, Antonio	TU	7631	424	agav@sip.ucm.es

Figura 91: Hoja de cálculo de los profesores

7.3.3. Elección de docencia

El proceso de elección de docencia se realiza a través de los *endpoints* de la tabla 5

Endpoint	GET	POST	PUT	PATCH	DELETE
/api/docencias/	X	X			
/api/docencias/comprobar/		X			
/api/docencias/reiniciar/	X				
/api/docencias/{id}/	X		X	X	X
/api/docencias/{docencia_pk}/divisibles/	X	X			
/api/docencias/{docencia_pk}/divisibles/{id}/	X		X	X	X
/api/docencias/incompatibles/		X			

Tabla 5: Endpoints API - Elección de docencia

Algunos de los *endpoints* más destacables son:

- El *endpoint* /api/docencias/comprobar/ permite simular una elección de docencia y comprobar si es válida y no viola ninguna de las restricciones. Es útil para que la aplicación web informe durante la elección si las asignaturas que se eligen son compatibles entre sí.
- El *endpoint* /api/docencias/reiniciar/ elimina todas las elecciones que existan en el sistema y restaura las variables del departamento que controlan el proceso de elección a sus valores iniciales.
- El *endpoint* /api/docencias/incompatibles/ permite conocer a través de la entrada de una serie de asignaturas y desdobles, que asignaturas no son compatibles debido a solapamiento de horarios.

A continuación, se muestra un ejemplo de la respuesta que se recibe al comprobar una elección de docencia con incompatibilidades:

```
POST /api/docencias/comprobar/
HTTP 200 OK
Allow: POST, OPTIONS
Content-Type: application/json
Vary: Accept
{
  "M": {
    "primer_cuatrimestre": [
      [
        {
```

```

        "nombre": "Bases de datos",
        "siglas": "BD",
        "curso": "2º",
        "grupo": "I",
        "asignatura": 6,
        "horario_solapado": [
            "15:00:00",
            "16:50:00"
        ],
        "titulacion": "GI",
        "codigo": "803273",
        "calendario": "Grados y máster [2018-09-12 - 2018-12-20]"
    },
    {
        "nombre": "Aplicaciones web",
        "siglas": "AW",
        "curso": "4º",
        "grupo": "E",
        "asignatura": 3,
        "horario_solapado": [
            "16:00:00",
            "17:50:00"
        ],
        "titulacion": "GS",
        "codigo": "803337",
        "calendario": "Grados y máster [2018-09-12 - 2018-12-20]"
    }
]
},
"segundo_cuatrimestre": []
},
"x": null,
"l": null,
"v": null,
"j": null
}

```

Para crear una docencia con asignaturas divisibles y no divisibles se podría enviar una petición como la del ejemplo:

```

POST /api/docencias/
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
  "asignaturas": [8, 9],
  "desdobles": [],
  "asignaturas_divisibles": [
    {"asignatura": 3, "creditos": 2.5}
  ],
  "profesor": 3
}

```

Cuando un profesor elige docencia es necesario modificar algunas variables del departamento: la capacidad restante, el número de créditos asignados, la carga dinámica, etc. La creación de una nueva docencia supone la variación de la carga dinámica del departamento, que como se explica en el apartado 7.3.4, resulta de la división de los créditos que quedan pendientes de asignar y de la capacidad restante del departamento. Además, con la variación de la carga dinámica también varía el número de créditos que debe escoger cada profesor en su elección. En la figura 92 se muestra el fragmento de código que actualiza estos valores.

```

"""
Actualiza los créditos del departamento cuando se crea o se actualiza una docencia
"""
@receiver(pre_save, sender=EleccionDocencia)
def my_handler_save(sender, **kwargs):
    departamento = Departamento.objects.get(siglas=kwargs['instance'].profesor.departamento)

    eleccion = kwargs['instance']
    profesor = kwargs['instance'].profesor

    # Si la elección se está actualizando hay que restar los créditos anteriores
    if eleccion.id:
        old_eleccion = EleccionDocencia.objects.get(pk=eleccion.id)
        creditos = profesor.pda + old_eleccion.deuda
        creditos = -creditos
        departamento.capacidad_restante += old_eleccion.profesor.categoria.capacidad_docente

    departamento.capacidad_restante -= profesor.categoria.capacidad_docente
    creditos = profesor.pda + eleccion.deuda
    departamento.update_creditos(creditos, profesor)

```

Figura 92: Fragmento de código - Actualización variables departamento

Para la comprobación de solapamientos entre asignaturas, el algoritmo implementado crea una lista de intervalos horarios por cada cuatrimestre, y posteriormente se calculan los solapamientos que existen entre los intervalos como se muestra en la figura 93.

```

overlapping = {"primer_cuatrimestre": list(
), "segundo_cuatrimestre": list()}

# Solapamientos primer cuatrimestre
[overlapping["primer_cuatrimestre"].append([x, y]) for x in intervals_primer_c for y in intervals_primer_c
 | if x is not y and x["horario_solapado"][1] >= y["horario_solapado"][0] and x["horario_solapado"][0]
 | <= y["horario_solapado"][0] and [y, x] not in overlapping["primer_cuatrimestre"]
 | or x["horario_solapado"][0] = y["horario_solapado"][0] and x["horario_solapado"][1] = y["horario_solapado"][1]
 | and [y, x] not in overlapping["primer_cuatrimestre"] and x is not y]

# Solapamientos segundo cuatrimestre
[overlapping["segundo_cuatrimestre"].append([x, y]) for x in intervals_segundo_c for y in intervals_segundo_c
 | if x is not y and x["horario_solapado"][1] >= y["horario_solapado"][0] and x["horario_solapado"][0]
 | <= y["horario_solapado"][0] and [y, x] not in overlapping["segundo_cuatrimestre"]
 | or x["horario_solapado"][0] = y["horario_solapado"][0] and x["horario_solapado"][1] = y["horario_solapado"][1]
 | and [y, x] not in overlapping["segundo_cuatrimestre"] and x is not y]

if len(overlapping["primer_cuatrimestre"]) + len(overlapping["segundo_cuatrimestre"]) = 0:
 | overlapping = None

return overlapping

```

Figura 93: Fragmento de código - Comprobación solapamientos entre asignaturas

Finalmente, la figura 94 muestra el algoritmo utilizado para comprobar que asignaturas son incompatibles con una asignatura dada debido al solapamiento de horas.

```

@staticmethod
def incompatibilidad(dia, asignatura, asignaturas):
    incompatibles = dict()

    for a in asignaturas[dia]:
        intervals = asignatura["horario"][dia] + asignaturas[dia][a]

        intervals.sort()
        l = len(intervals)
        overlaps = list()
        for i in range(l):
            for j in range(i+1, l):
                x = intervals[i]
                y = intervals[j]
                if x[0] = y[0]:
                    overlaps.append([x, y])
                elif x[1] = y[1]:
                    overlaps.append([x, y])
                elif (x[1] > y[0] and x[0] < y[0]):
                    overlaps.append([x, y])

            if len(overlaps) > 0:
                incompatibles[a] = overlaps

    return incompatibles

```

Figura 94: Fragmento de código - Incompatibilidades con asignatura

7.3.4. Departamentos

La aplicación de departamentos, a diferencia de las anteriores, no consta de la misma complejidad. Sin embargo, es totalmente necesaria para mantener un control sobre las variables que condicionan el proceso de elección.

La carga dinámica del departamento determina el número de créditos que deben escoger los profesores durante la elección. Este valor resulta de la división de los créditos que quedan pendientes de asignar y de la capacidad restante del departamento. La capacidad restante es la diferencia de los créditos de docencia, deudas y PDA menos los créditos fijos del personal asociado, becario y ayudante. Ésta se va actualizando según avanza el proceso restando la capacidad docente de cada profesor.

Finalmente, el número de créditos que debe elegir cada profesor se calcula mediante la multiplicación de la carga dinámica del departamento por la capacidad docente del profesor.

La cantidad de *endpoints* disponibles es menor a los de las aplicaciones anteriores debido a que no es habitual realizar operaciones CRUD en este módulo, ya que la actualización de los recursos se realiza automáticamente (ver tabla 6).

Endpoint	GET	POST	PUT	PATCH	DELETE
/api/departamentos/	X	X			
/api/departamentos/{siglas}/	X		X	X	X

Tabla 6: Endpoints API - Departamentos

A continuación, se muestra un ejemplo de la respuesta que se recibe al solicitar un departamento concreto.

```
GET /api/departamentos/SIC/
```

```
HTTP 200 OK
```

```
Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS
```

```
Content-Type: application/json
```

```
Vary: Accept
```

```
{  
  "siglas": "SIC",  
  "nombre": "Sistemas Informáticos y Computación",  
  "total_creditos": 706.5,  
  "creditos_doble": 132.0,  
  "deudas": 0.0,  
  "deudas_corregidas": 0.0,
```

```
"pda": 0.0,  
"total_creditos_carga": 706.5,  
"creditos_fijos": 195.0,  
"resto_creditos": 511.5,  
"creditos_asignados": 41.5,  
"creditos_sin_asignar": 470.0,  
"creditos_dobles_sin_asignar": 129.0,  
"capacidad_restante": 463.5,  
"carga_dinamica": 1.0140237324703345,  
"docencia_iniciada": true  
}
```

En la figura 95 se muestran los dos métodos que calculan y actualizan las variables del departamento y modifican el valor de los créditos que deben elegir los profesores en sus docencias. Cada vez que se registra una nueva elección, se deben actualizar estos valores, ya que al disminuir los créditos que quedan por asignar en el departamento, la carga dinámica varía y ésta afecta directamente al número de créditos que deben escoger todos los profesores.

```

def calcular(self):
    self.total_creditos_carga = self.total_creditos + \
        self.deudas_corregidas + self.pda
    self.resto_creditos = self.total_creditos_carga - self.creditos_fijos

    if not self.docencia_iniciada:
        self.creditos_asignados = 0
        self.creditos_sin_asignar = self.resto_creditos
        self.creditos_desdobles_sin_asignar = self.creditos_desdoble
        self.capacidad_restante = self.resto_creditos
        self.carga_dinamica = 1

def update_creditos(self, creditos, profesor):
    from profesores.models import Profesor

    self.creditos_asignados += creditos
    self.creditos_sin_asignar -= creditos

    self.carga_dinamica = self.creditos_sin_asignar / \
        self.capacidad_restante

    """
    Acumular créditos escogidos por el profesor (se llama 3 veces: para PDA y deuda,
    asignaturas y desdobles)
    """
    Profesor.objects.filter(usuario=profesor.usuario).update(
        creditos_escogidos=F('creditos_escogidos') + creditos)

    """
    Actualización de los créditos que deben escoger los demás profesores
    """
    profesores = Profesor.objects.update(creditos_escoger=Subquery(
        Profesor.objects.filter(
            usuario=OuterRef('usuario')
        ).annotate(
            creditos=F('categoria__capacidad_docente') *
            self.carga_dinamica
        ).values('creditos')
    ))

    self.save()

```

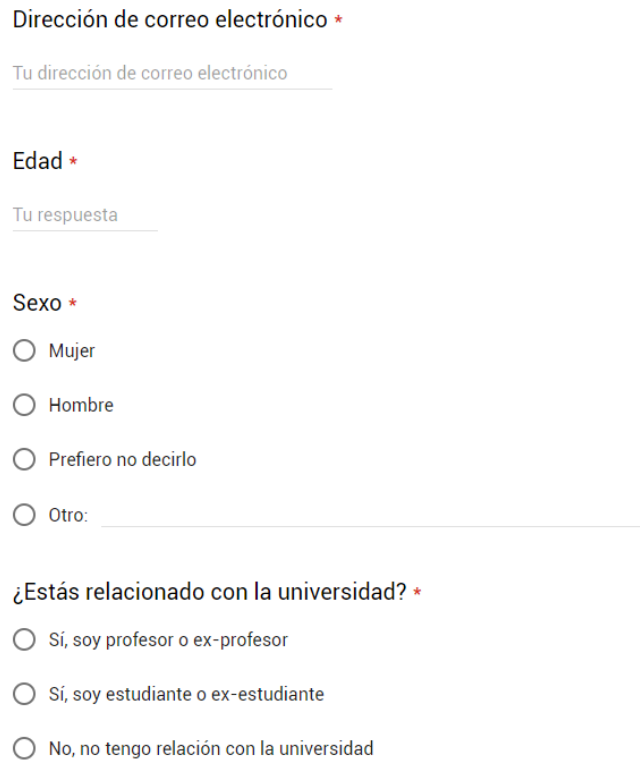
Figura 95: Fragmento de código - Cálculo de variables del departamento y actualización de los créditos de los profesores

8. EVALUACIÓN

En este capítulo se describirán los resultados de una evaluación que se ha realizado sobre la aplicación desarrollada. Se han buscado personas con tres niveles diferentes de conexión con el mundo universitario: profesores o ex profesores, estudiantes o ex estudiantes, y personas ajenas a la universidad. No se ha acotado la franja de edades.

La evaluación se ha realizado con Google Forms y se ha dividido en cuatro bloques:

1. Detalles generales del encuestado (ver figura 96).



Dirección de correo electrónico *

Tu dirección de correo electrónico

Edad *

Tu respuesta

Sexo *

Mujer

Hombre

Prefiero no decirlo

Otro: _____

¿Estás relacionado con la universidad? *

Sí, soy profesor o ex-profesor

Sí, soy estudiante o ex-estudiante

No, no tengo relación con la universidad

Figura 96: Formulario - Detalles generales del encuestado

2. Cuestionario de satisfacción enfocado a las pruebas de la aplicación desde el punto de vista de un profesor (ver figuras 97, 98, 99 y 100).

1. Haz login con una cuenta de profesor

Todos los usuarios tienen como nombre de usuario un correo electrónico con dominio "sip.ucm.es" y como contraseña "profesor".

Ejemplo:

Nombre de usuario: <correo>@sip.ucm.es
Contraseña: profesor

Indica tu nivel de satisfacción con la pantalla de login

0 1 2 3 4 5 6 7 8 9 10
Muy bajo Muy alto

Indica tu nivel de satisfacción con la pantalla de dashboard (la de las gráficas)

0 1 2 3 4 5 6 7 8 9 10
Muy bajo Muy alto

Indica tu nivel de satisfacción con el menú

0 1 2 3 4 5 6 7 8 9 10
Muy bajo Muy alto

Figura 100: Formulario - Pruebas desde el punto de vista de un profesor I

5. En el menú lateral accede al listado de profesores

Indica tu nivel de satisfacción con el listado de profesores

0 1 2 3 4 5 6 7 8 9 10
Muy bajo Muy alto

6. En las opciones de la derecha o el campo de texto de encima de la lista, realiza algún filtro

Por ejemplo: filtra por apellido o por categoría

Indica tu nivel de satisfacción con el sistema de filtrado

0 1 2 3 4 5 6 7 8 9 10
Muy bajo Muy alto

Figura 98: Formulario - Pruebas desde el punto de vista de un profesor III

2. En el menú lateral accede al listado de asignaturas

Indica tu nivel de satisfacción con el listado de asignaturas

0 1 2 3 4 5 6 7 8 9 10
Muy bajo Muy alto

3. En las opciones de la derecha o el campo de texto de encima de la lista, realiza algún filtro

Por ejemplo: filtra por siglas o por cuatrimestre

Indica tu nivel de satisfacción con el sistema de filtrado

0 1 2 3 4 5 6 7 8 9 10
Muy bajo Muy alto

4. Escoge una de ellas y fíjate en los detalles

Indica tu nivel de satisfacción con la página de detalle de la asignatura

0 1 2 3 4 5 6 7 8 9 10
Muy bajo Muy alto

Figura 97: Formulario - Pruebas desde el punto de vista de un profesor II

7. Escoge uno de ellos y fíjate en los detalles

Indica tu nivel de satisfacción con la página de detalle del profesor

0 1 2 3 4 5 6 7 8 9 10
Muy bajo Muy alto

8. En el menú lateral, accede a la sección "Elección de docencia"

Indica tu nivel de satisfacción con la página de la elección de docencia

0 1 2 3 4 5 6 7 8 9 10
Muy bajo Muy alto

9. Elige asignaturas y fíjate en el calendario

Indica tu nivel de satisfacción con el sistema de elección de asignaturas y el diseño del calendario

0 1 2 3 4 5 6 7 8 9 10
Muy bajo Muy alto

Figura 99: Formulario - Pruebas desde el punto de vista de un profesor IV

3. Cuestionario de satisfacción enfocado a las pruebas de la aplicación desde el punto de vista de un administrador (ver figura 101).

Encuesta de satisfacción del Sistema de elección de docencia

1. Haz login con una cuenta de administrador

Nombre de usuario: jpicas@ucm.es
Contraseña: profesoradministrador

2. En el menú lateral accede al listado de asignaturas y crea una asignatura nueva o edita una existente

Indica tu nivel de satisfacción con la el formulario de creación o edición de una asignatura

0 1 2 3 4 5 6 7 8 9 10

Muy bajo Muy alto

3. En el menú lateral accede al listado de profesores y crea un profesor nuevo o edita uno existente

Indica tu nivel de satisfacción con la el formulario de creación o edición de un profesor

0 1 2 3 4 5 6 7 8 9 10

Muy bajo Muy alto

4. En el menú lateral accede al sistema de elección de docencia y navega a través de las elecciones de los diferentes profesores

Indica tu nivel de satisfacción con el sistema de elección de docencia

0 1 2 3 4 5 6 7 8 9 10

Muy bajo Muy alto

Figura 101: Formulario - Pruebas desde el punto de vista de un administrador

4. Nivel de satisfacción general del diseño y la fluidez y velocidad de la aplicación (ver figura 102).

Encuesta de satisfacción del Sistema de elección de docencia

En general, ¿cual es tu nivel de satisfacción con el diseño?

0 1 2 3 4 5 6 7 8 9 10

Muy malo Muy Bueno

En general, ¿cual es tu nivel de satisfacción con la velocidad y fluidez de la aplicación?

0 1 2 3 4 5 6 7 8 9 10

Muy malo Muy Bueno

¿Quieres decirnos algo? Todas las sugerencias serán bienvenidas

Tu respuesta _____

Figura 102: Formulario - Satisfacción diseño, fluidez y velocidad de la aplicación

A continuación, se listan los resultados de la encuesta divididos en los grupos anteriormente mencionados.

8.1. Detalles generales del encuestado

En la figura 103 se observa que la edad de los usuarios encuestados es muy variada. El 47% de los encuestados tienen una edad comprendida entre los 18 y 22 años. Mientras que el 53% restante tiene una edad comprendida entre los 30 y 53 años.

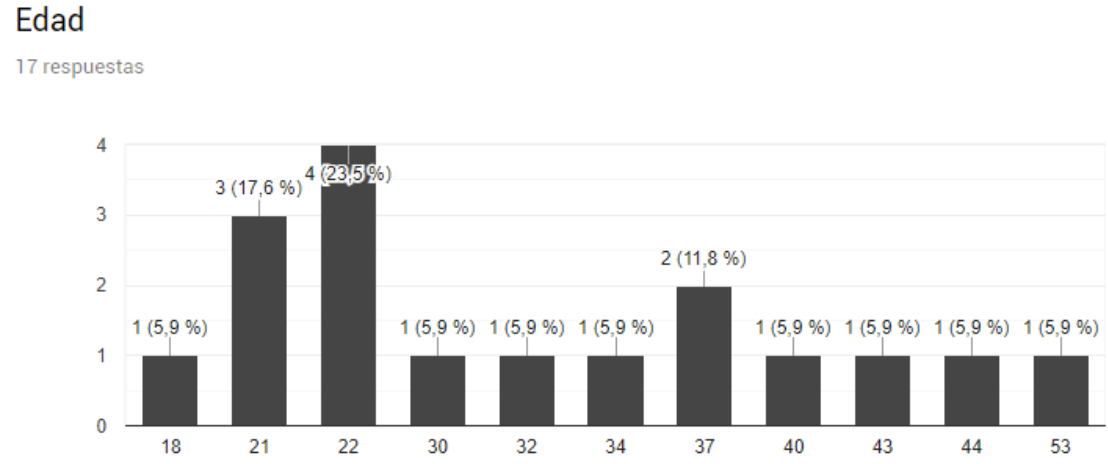


Figura 103: Evaluación - Edad de los encuestados

En cuanto al sexo de los encuestados predomina el masculino con un 82,4%, mientras que el 17,6% corresponde al femenino (ver figura 104).

Sexo

17 respuestas

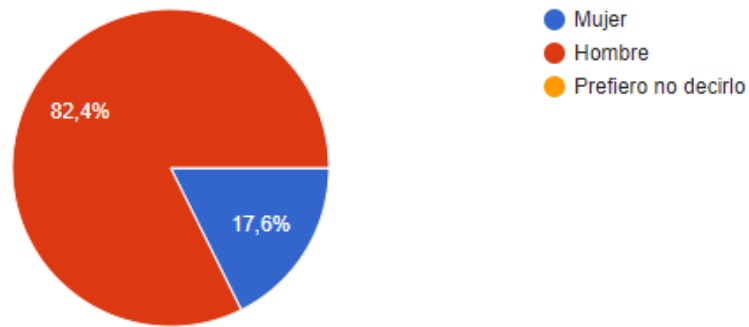


Figura 104: Evaluación - Sexo de los encuestados

El 88,2% de los encuestados tienen o han tenido alguna relación con el ámbito universitario (ver figura 105).

¿Estás relacionado con la universidad?

17 respuestas

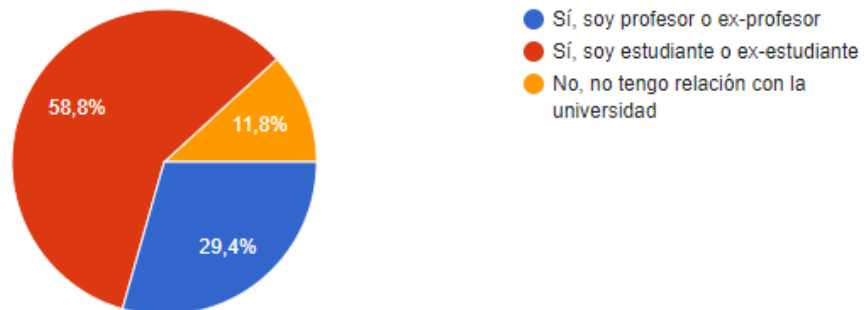


Figura 105: Evaluación - Relación con la universidad del encuestado

8.2. Cuestionario de satisfacción desde el punto de vista de un profesor

En cuanto a la satisfacción con la pantalla de *login* el 82,3% de los encuestados le otorgan una nota superior o igual a 8 (ver figura 106).

Indica tu nivel de satisfacción con la pantalla de login

17 respuestas

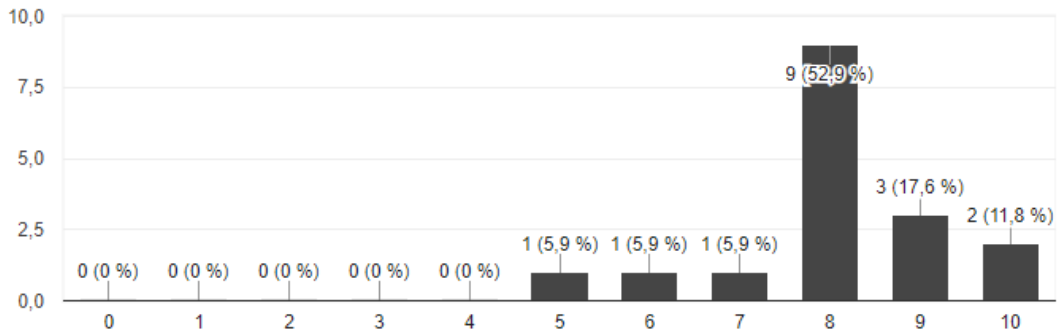


Figura 106: Evaluación - Profesor - Pantalla de login

La pantalla de *dashboard* que contiene el resumen gráfico del estado de la elección de docencia ha sido valorada por el 82,4% de los encuestados con una nota de entre 8 y 9 (ver figura 107).

Indica tu nivel de satisfacción con la pantalla de dashboard (la de las gráficas)

17 respuestas

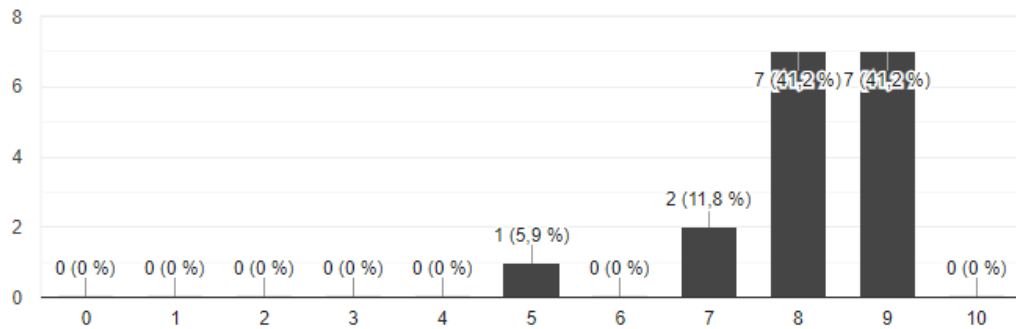


Figura 107: Evaluación - Profesor - Pantalla de dashboard

El nivel de satisfacción con el menú se valoró con una nota de entre 7 y 10 por el 100% de los encuestados (ver figura 108).

Indica tu nivel de satisfacción con el menú

17 respuestas

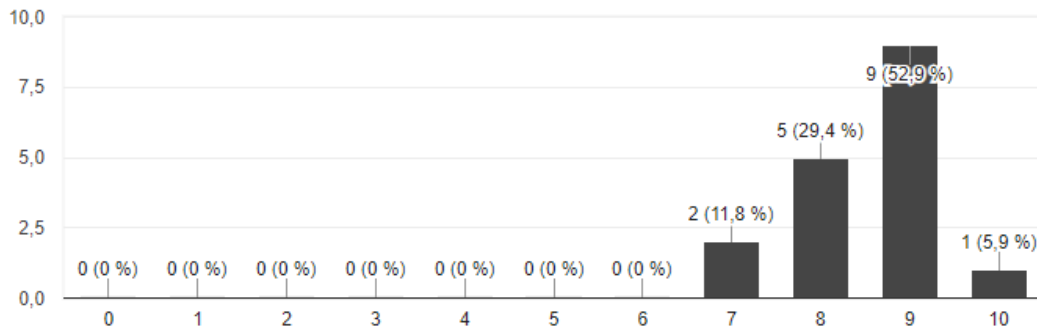


Figura 108: Evaluación - Asignatura - Menú

En cuanto al listado de asignaturas, el 76,5% de los encuestados lo valoraron con una nota de entre 8 y 10 (ver figura 109).

Indica tu nivel de satisfacción con el listado de asignaturas

17 respuestas

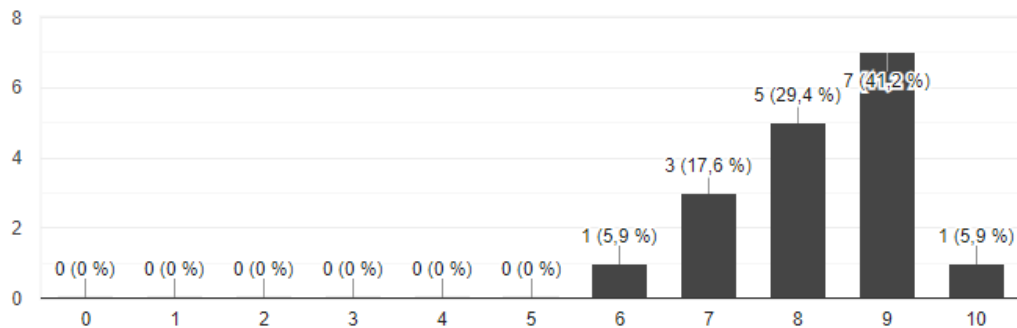


Figura 109: Evaluación - Profesor - Listado de asignaturas

El sistema de filtrado de asignaturas se valoró por el 82,3% de los encuestados con una nota de entre 8 y 10 (ver figura 110).

Indica tu nivel de satisfacción con el sistema de filtrado

17 respuestas

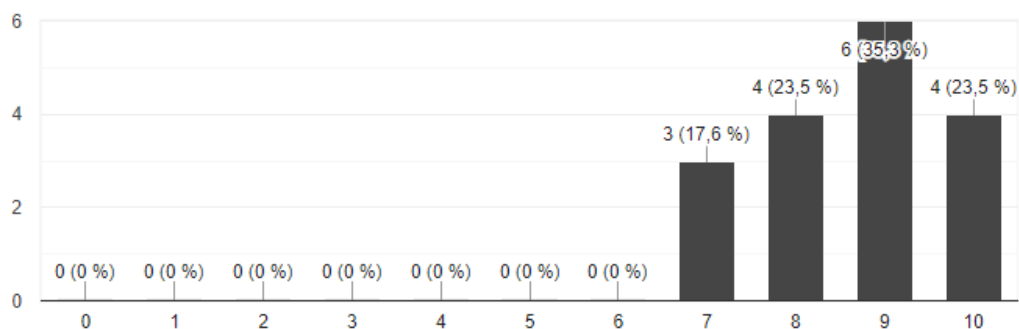


Figura 110: Evaluación - Profesor - Filtrado de asignaturas

El 29,4% de los encuestados valoró la página de detalle de las asignaturas con una nota de entre 6 y 7. El 70,6% restante lo valoró con una nota superior a 7 (ver figura 111).

Indica tu nivel de satisfacción con la página de detalle de la asignatura

17 respuestas

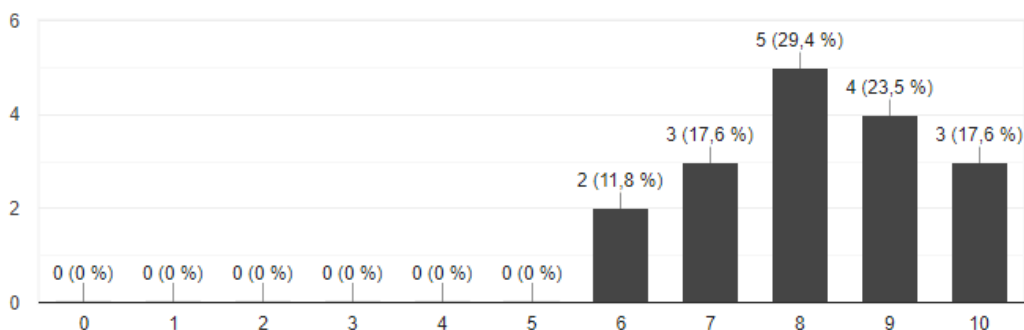


Figura 111: Evaluación - Profesor - Detalle de la asignatura

En cuanto al listado de profesores, el 88,3% de los usuarios lo valoró con una nota de entre 8 y 10 (ver figura 112).

Indica tu nivel de satisfacción con el listado de profesores

17 respuestas

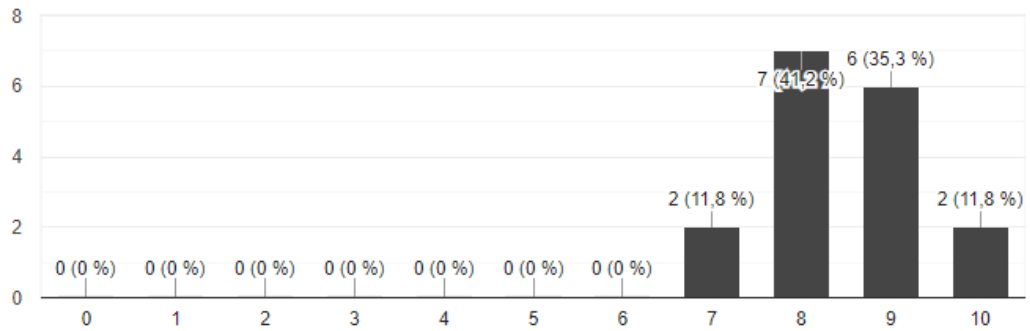


Figura 112: Evaluación - Profesor - Listado de profesores

El sistema de filtrado de profesores se valoró por el 70,6% de los usuarios con una nota de entre 8 y 10, mientras que el 29,4% lo valoró con una nota de entre 6 y 7 (ver figura 113).

Indica tu nivel de satisfacción con el sistema de filtrado

17 respuestas

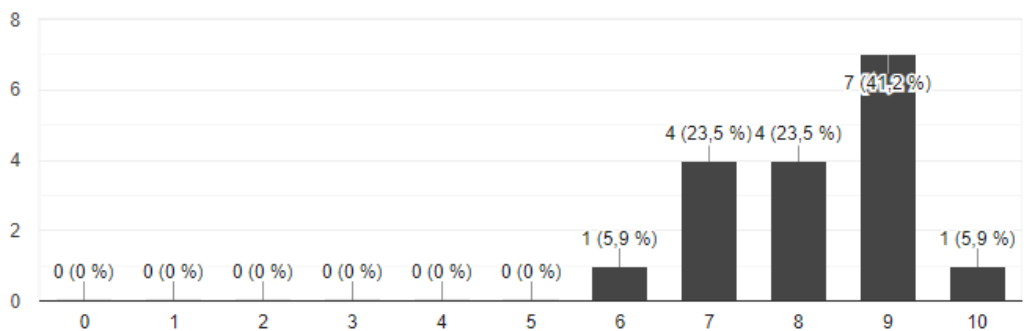


Figura 113: Evaluación - Profesor - Filtrado de profesores

La página de detalle del profesor recibió una nota de entre 7 y 9 por el 94% de los encuestados, el 5,9% lo valoró con una nota de 5 (ver figura 114).

Indica tu nivel de satisfacción con la página de detalle del profesor

17 respuestas

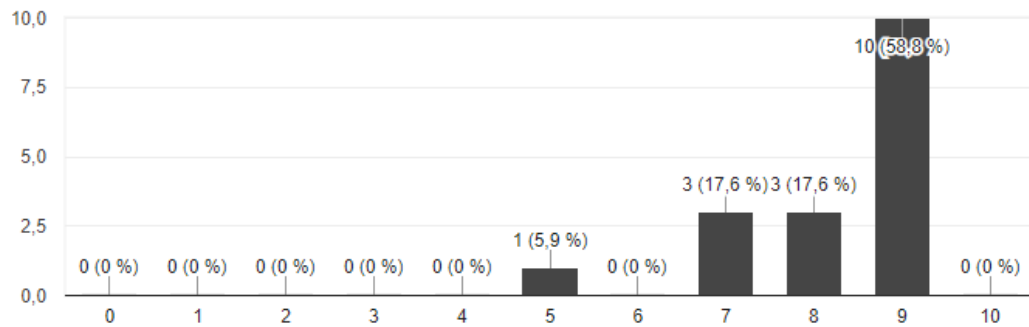


Figura 114: Evaluación - Profesor - Detalle del profesor

La página de elección de docencia fue valorada por el 100% de los encuestados con una nota de entre 8 y 10 (ver figura 115).

Indica tu nivel de satisfacción con la página de la elección de docencia

17 respuestas

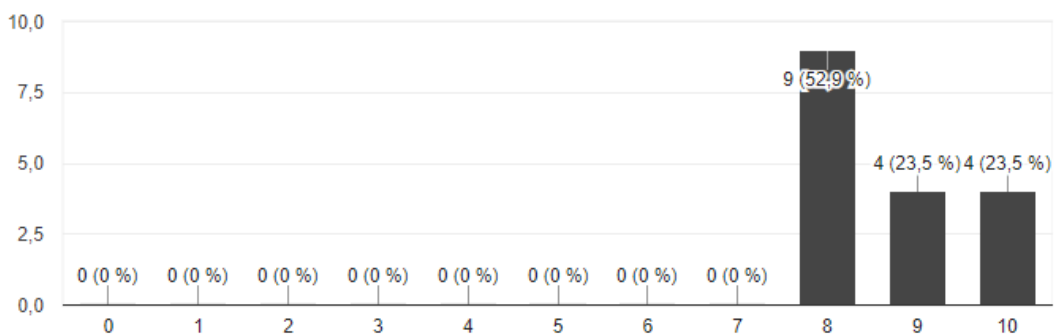


Figura 115: Evaluación - Profesor - Pantalla de elección de docencia

En cuanto al nivel de satisfacción con el sistema de elección de asignaturas y el diseño del calendario, el 94,1% de los encuestados lo valoró con una nota superior o igual a 8 (ver figura 116).

Indica tu nivel de satisfacción con el sistema de elección de asignaturas y el diseño del calendario

17 respuestas

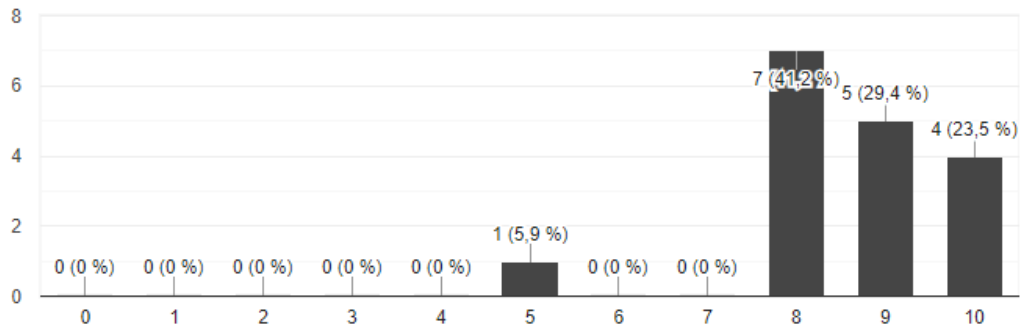


Figura 116: Evaluación - Profesor - Diseño del calendario de la elección de docencia

8.3. Cuestionario de satisfacción desde el punto de vista de un administrador

El nivel de satisfacción con el formulario de creación o edición de una asignatura ha sido valorado con una nota de entre 7 y 10 por el 88,9% de los encuestados, mientras que el 11,1% lo valoró con una nota de 2 o 5 (ver figura 117).

Indica tu nivel de satisfacción con el formulario de creación o edición de una asignatura

18 respuestas

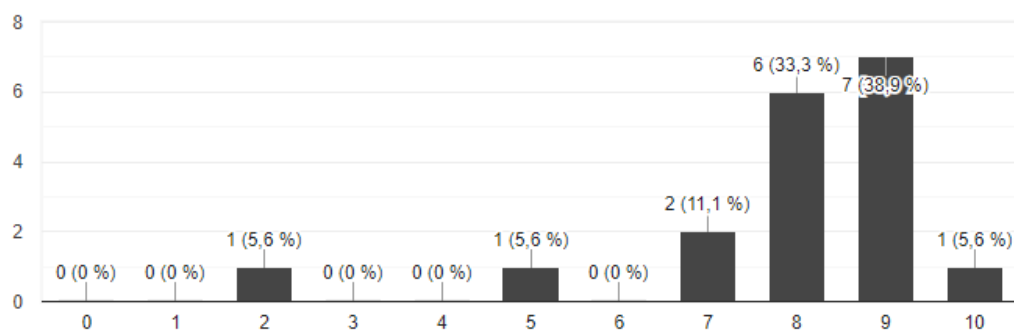


Figura 117: Evaluación - Administrador - Crear / Editar asignatura

El formulario de creación y de edición de profesores fue valorado por el 88,2% de los usuarios con una nota de entre 8 y 10, el 11,8% lo valoró con una nota de entre 5 y 6 (ver figura 118).

Indica tu nivel de satisfacción con la el formulario de creación o edición de un profesor

17 respuestas

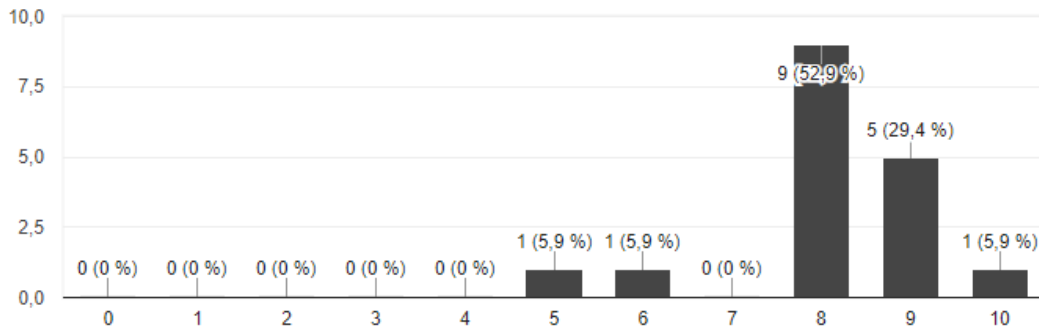


Figura 118: Evaluación - Administrador - Crear / Editar profesor

El sistema de elección de docencia fue valorado con una nota de entre 7 y 9 por el 94,1% de los encuestados (ver figura 119).

Indica tu nivel de satisfacción con el sistema de elección de docencia

17 respuestas

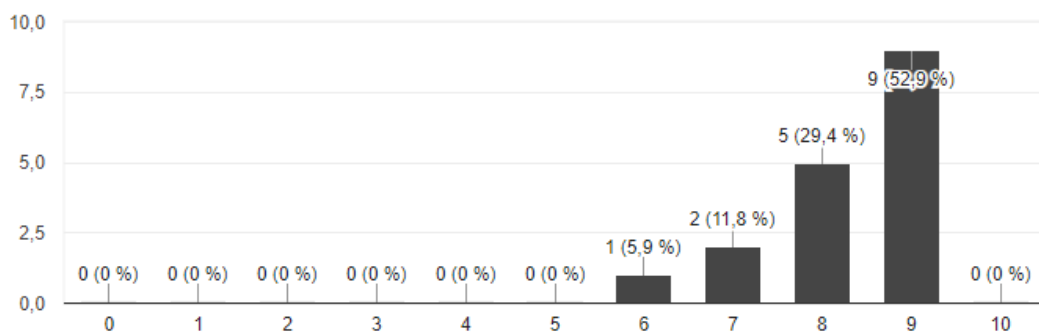


Figura 119: Evaluación - Administrador - Elección de docencia

8.4. Nivel de satisfacción general

El 94,1% de los usuarios encuestados otorgó una nota superior a 7 al diseño general de la aplicación web (ver figura 120).

En general, ¿cual es tu nivel de satisfacción con el diseño?

17 respuestas

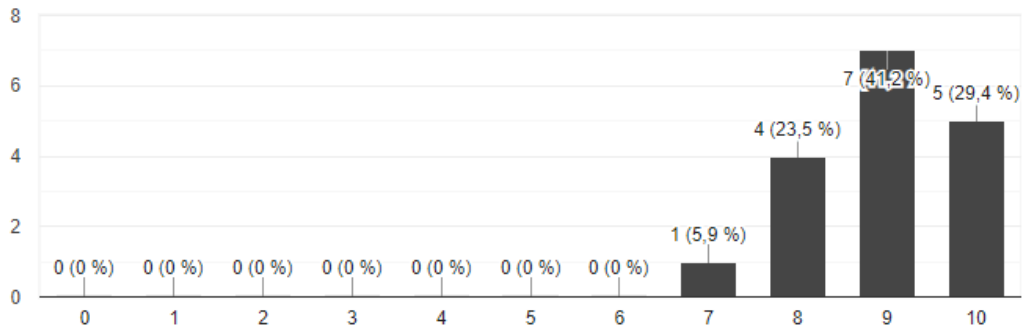


Figura 120: Evaluación – Diseño

La velocidad y fluidez de la aplicación fue valorada por el 76,4% de los encuestados con una nota de entre 8 y 10 (ver figura 121).

En general, ¿cual es tu nivel de satisfacción con la velocidad y fluidez de la aplicación?

17 respuestas

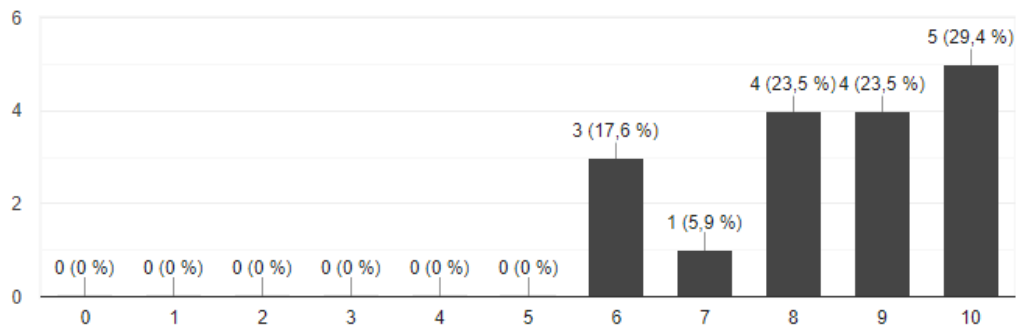


Figura 121: Evaluación - Velocidad y fluidez

9. CONCLUSIONES Y TRABAJO FUTURO

9.1. Conclusiones

En este trabajo se ha desarrollado una herramienta que permite a los profesores elegir la docencia a través de una aplicación web. Se ha procurado automatizar el mayor número de tareas. Mediante la lectura de hojas de cálculo, el sistema es capaz de arbitrar el proceso de elección de docencia con un menor grado de intervención por parte del coordinador. Sin embargo, la figura del coordinador sigue siendo necesaria debido a las variadas casuísticas que pueden aparecer durante el proceso.

Para realizar este trabajo se han utilizado tecnologías de código abierto para el desarrollo de aplicaciones web.

Durante el desarrollo de este trabajo se han puesto en práctica conceptos de ingeniería de software, bases de datos, desarrollo de sistemas interactivos y programación, entre otros.

9.2. Trabajo futuro

Aunque se han conseguido implementar la mayoría de las funcionalidades que se propusieron al inicio de este trabajo, hay varios aspectos que se plantean como líneas de trabajo futuro:

- **Sistema de autenticación.** En el sistema desarrollado las cuentas de usuario se gestionan en el mismo servidor que se aloja la aplicación. Esto obliga a los docentes a tener unas credenciales nuevas para identificarse en la aplicación. En consecuencia, una línea de trabajo es desarrollar un sistema de autenticación que delegue la gestión de cuentas en la propia intranet del departamento o centro al que pertenecen los profesores.
- **Gestión de reservas.** Los profesores tienen la posibilidad de reservar la docencia de asignaturas antes de iniciarse el proceso de elección. Sin embargo, la herramienta desarrollada no provee esta funcionalidad. Se trata de una tarea compleja. En este sentido, una mejora de la aplicación consiste en añadir esta funcionalidad debido a que es bastante común que los profesores reserven la docencia de las asignaturas que habitualmente imparten.
- **Histórico de docencias.** Mantener un histórico de las elecciones docentes pasadas sería de gran utilidad para estimar las elecciones futuras. De esta manera, los docentes podrían intuir qué asignaturas estarán disponibles cuando les llegue el turno de elección.
- **Mayor grado de automatización.** Para el correcto funcionamiento de la herramienta propuesta es necesaria la figura de un administrador que controle el desarrollo del proceso. El administrador debe intervenir cuando se den situaciones bloqueantes o excepcionales. Para reducir el grado de intervención por parte del administrador, una mejora consistiría en dotar a la herramienta de la inteligencia suficiente para tener en cuenta más restricciones y posibles soluciones a estas.
- **Gestión de personal para el cuidado de exámenes.** Aunque no está directamente relacionado con la gestión de docencia, sería de gran utilidad que la herramienta diese apoyo a la gestión de personal que ayuda en el cuidado de los exámenes.

- **Mejoras y correcciones.** El mantenimiento del software es una de las fases del ciclo de vida del desarrollo de sistemas. Mediante el *feedback* de los usuarios se espera corregir errores, mejorar las capacidades, eliminar funciones que hayan quedado obsoletas y optimizar la herramienta.

9. CONCLUSIONS AND FUTURE WORK

9.1. Conclusions

This work has developed a tool that allows teachers to choose teaching through a web application. An attempt has been made to automate as many tasks as possible. By reading spreadsheets, the system is able to arbitrate the process of choosing teachers with a lesser degree of intervention by the coordinator. However, the figure of the coordinator is still necessary due to the variety of cases that may appear during the process.

In order to carry out this work, open source technologies have been used for the development of web applications.

During the development of this work have been put into practice concepts of software engineering, databases, development of interactive systems and programming, among others.

9.2. Future work

Although most of the functionalities proposed at the beginning of this work have been implemented, there are several aspects that are proposed as future lines of work:

- **Authentication system.** In the developed system the user accounts are managed in the same server that hosts the application. This requires teachers to have new credentials to identify themselves in the application. Consequently, one line of work is to develop an authentication system that delegates the management of accounts to the intranet of the department or centre to which the teachers belong.
- **Reservations management.** Teachers have the possibility of reserving the teaching of subjects before the election process begins. However, the tool developed does not provide this functionality. This is a complex task. In this sense, an improvement of the application consists of adding this functionality because it is quite common for teachers to reserve the teaching of the subjects they usually teach.
- **History of teaching.** Maintaining a history of past teacher elections would be very useful in estimating future elections. In this way, teachers could sense what subjects will be available when it is their turn to choose.
- **Greater degree of automation.** For the proper functioning of the proposed tool is necessary the figure of an administrator to control the development of the process. The administrator must intervene when there are blocking or exceptional situations. In order to reduce the degree of intervention on the part of the administrator, an improvement would consist of providing the tool with sufficient intelligence to take into account more restrictions and possible solutions to them.
- **Management of personnel for the care of examinations.** Although it is not directly related to the management of teaching, it would be very useful if the tool were to support the management of personnel who assist in the care of examinations.

- **Improvements and corrections.** Software maintenance is one of the life cycle phases of system development. User feedback is expected to correct errors, improve capabilities, remove obsolete functions and optimize the tool.

10. APORTACIONES INDIVIDUALES

En este capítulo se explicarán qué tareas ha realizado cada alumno durante el desarrollo del proyecto.

10.1. David Arroyo Segovia

En cuanto a la contribución de David al proyecto se distinguen los siguientes apartados:

Ingeniería de requisitos.

Para abordar este trabajo fue necesario dedicar una gran parte del tiempo a la ingeniería de requisitos. No se conocía en profundidad el funcionamiento del proceso de elección de docencia que debía gestionar la herramienta que se ha desarrollado. En este sentido, fue necesario realizar reuniones en las que se tuvo que determinar las necesidades y condiciones a satisfacer por el software.

Modelado de los datos.

Para realizar el modelado de los datos, junto con Javier, se realizaron distintos diagramas conceptuales basándose en los requisitos que se iban capturando en las reuniones sucesivas con los directores de este trabajo. En las primeras fases de desarrollo, el enfoque del modelado era de bastante alto nivel, pero a medida que avanzaba el desarrollo este iba tomando un enfoque de más bajo nivel. El modelo de datos fue una tarea que se realizó en conjunto debido a que influía directamente en la implementación de cada parte de la arquitectura, tanto de la aplicación web como del diseño de la API.

Diseño e implementación de la API REST.

Se diseñó y se implementó una API REST con Django. Se establecieron una serie de reglas para comunicar el servicio REST con la aplicación web. En consecuencia, se decidieron los servicios que ofrecería la API, la forma de consumirlos y de acceder a ellos. Las partes más significativas que se implementaron fueron:

- El modelo de datos en Django.
- Los *serializadores* y vistas de los distintos recursos representables por la API.
- La importación de asignaturas y profesores a partir de hojas de cálculo.
- La comprobación de restricciones en las elecciones docentes.
- El sistema de autenticación.

Sistema de autenticación basado en JSON Web Token.

Durante el desarrollo del trabajo se planteó la posibilidad de delegar la autenticación en la propia intranet del departamento al que pertenecen nuestros directores. Sin embargo, no fue posible debido a inconvenientes encontrados durante la conexión de ambos sistemas. Debido a esto, se buscó una solución para autenticar a los usuarios. Se hizo uso de los JSON Web Token [41], un mecanismo de autenticación bastante extendido cuando se utilizan APIs.

Se implementó el sistema de autenticación tanto en la API como en la aplicación web (desarrollada con Angular).

Importación y exportación de profesores y asignaturas en la aplicación web.

Se implementó la funcionalidad de importar/exportar listados de profesores y asignaturas tanto en la API como en la aplicación web. Se añadieron las vistas correspondientes y se programó la parte relativa a la comunicación con el servicio REST.

Docencia de los profesores en la aplicación web.

Se añadió en la vista de detalle de un profesor un resumen de las asignaturas incluidas en su docencia. En este sentido, se aprendió a utilizar el *framework* Angular mediante cursos online y la ayuda de Javier.

Corrección de errores.

Se corrigieron los errores detectados durante la fase de pruebas y de evaluación con usuarios. Estos errores afectaban tanto a la API como a la aplicación web.

Despliegue de la API.

Se desplegó la API en un servidor Linux para que estuviera operativa indefinidamente. En el apartado 2 del anexo II se puede encontrar una guía de instalación.

Memoria.

La redacción de la memoria se realizó junto con Javier y el apoyo de los directores de este Trabajo de Fin de Grado.

10.2. Javier Picatoste Zangróniz

En cuanto a la contribución de Javier a este trabajo se distinguen los siguientes apartados:

Ingeniería de requisitos y selección de tecnologías.

Durante las sucesivas reuniones se capturaron los requisitos necesarios y restricciones que debía satisfacer la aplicación. Tras un proceso de investigación se decidió utilizar Angular 6 para el *frontend* y Django para la API.

Se estudió la documentación oficial y varios tutoriales básicos sobre Angular 6. Se tomó la decisión de seguir las directrices de Material Design, normativa de diseño estudiada en la asignatura de Interfaces de Usuario.

Diseño e implementación de la capa de presentación.

Una vez adquiridos los conocimientos básicos y de integración de la librería de diseño, se configuró el entorno básico y se comenzó a implementar la capa de presentación utilizando Angular.

Las partes más significativas que se implementaron fueron:

- Los servicios que comunican la capa de presentación con la API REST.
- El código HTML y CSS de las distintas vistas de la aplicación.
- Los controladores de la aplicación.
- La *customización* de las librerías Chart.js y FullCalendar.
- Control de errores HTTP.

Antes de que se finalizará la implementación de la capa de presentación, David ya había terminado su funcionalidad principal así que se le explicaron los fundamentos de Angular 6.

Además, se confeccionó el logotipo de la aplicación utilizando los colores y tipografía oficial de la Complutense.

Despliegue de la aplicación web.

Una vez implementadas las operaciones CRUD, es decir, añadir, editar, ver y eliminar asignaturas, profesores y docencias se subió la aplicación a Heroku [42], un servicio gratuito online que permite que la aplicación sea accesible desde cualquier lugar y esté vinculada a un repositorio de GitHub.

Adaptación de la representación de recursos en la API.

En ocasiones fue necesario adaptar la representación de los recursos que se recibían en la capa de presentación desde la API. En este sentido, se modificó la API para que los recursos recibidos en formato JSON se adaptaran de una manera óptima a los tipos de datos utilizados en Angular.

Corrección de errores.

Una vez terminada la aplicación, se arreglaron los errores que se fueron detectando con la ayuda de la herramienta Trello (software de administración de proyectos) [43].

Memoria.

La redacción de la memoria se realizó en conjunto con David y el apoyo continuo de los directores de este Trabajo de Fin de Grado.

BIBLIOGRAFÍA

- [1] «Ellucian Banner Student». [En línea]. Disponible en: <https://www.ellucian.com/solutions/ellucian-banner-student>
- [2] «University and College Management Software». [En línea]. Disponible en: <http://www.ifnoss.com/edu/college-university-software/management-erp-system.aspx>
- [3] «Comprehensive University Timetabling System». [En línea]. Disponible en: <https://www.unitime.org/>
- [4] «Scheduling Software for School and University Timetables». [En línea]. Disponible en: <http://www.mimosasoftware.com/>
- [5] «Django REST framework». [En línea]. Disponible en: <https://www.django-rest-framework.org/>
- [6] «Django fundamental philosophies». [En línea]. Disponible en: <https://docs.djangoproject.com/en/2.2/misc/design-philosophies/>
- [7] «Librería openpyxl». [En línea]. Disponible en: <https://openpyxl.readthedocs.io/en/stable/>
- [8] «Python Virtual Environment». [En línea]. Disponible en: [https://wiki.archlinux.org/index.php/Python/Virtual_environment_\(Español\)](https://wiki.archlinux.org/index.php/Python/Virtual_environment_(Español))
- [9] «Angular». [En línea]. Disponible en: <https://angular.io/>
- [10] «Model–view–viewmodel - Implementations». [En línea]. Disponible en: <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93viewmodel#Implementations>
- [11] «HTML5». [En línea]. Disponible en: <https://developer.mozilla.org/es/docs/HTML/HTML5>
- [12] «HTML». [En línea]. Disponible en: <https://developer.mozilla.org/es/docs/Web/HTML>
- [13] «CSS3». [En línea]. Disponible en: <https://developer.mozilla.org/es/docs/Web/CSS>
- [14] «Sass documentation». [En línea]. Disponible en: <https://sass-lang.com/>
- [15] «Sass». [En línea]. Disponible en: [https://es.wikipedia.org/wiki/Sass_\(lenguaje_de_hojas_de_estilo\)](https://es.wikipedia.org/wiki/Sass_(lenguaje_de_hojas_de_estilo))
- [16] «Node Package Manager». [En línea]. Disponible en: [https://en.wikipedia.org/wiki/Npm_\(software\)](https://en.wikipedia.org/wiki/Npm_(software))
- [17] «What Is SQLite?» [En línea]. Disponible en: <https://www.sqlite.org/draft/index.html>
- [18] «SQLite». [En línea]. Disponible en: <https://es.wikipedia.org/wiki/SQLite>
- [19] «Apache HTTP Server». [En línea]. Disponible en: https://en.wikipedia.org/wiki/Apache_HTTP_Server

- [20] «Git». [En línea]. Disponible en: <https://git-scm.com/about>
- [21] «Visual Studio Code». [En línea]. Disponible en: https://es.wikipedia.org/wiki/Visual_Studio_Code
- [22] «VS Code extensions for near IDE experience». [En línea]. Disponible en: <https://medium.com/@spp020/vs-code-extensions-for-complete-ide-experience-bca5bb2f0f90>
- [23] «Gestión de contraseñas en Django». [En línea]. Disponible en: <https://docs.djangoproject.com/en/2.2/topics/auth/passwords/>
- [24] «El modelo cliente-servidor». [En línea]. Disponible en: <http://neo.lcc.uma.es/evirtual/cdd/tutorial/aplicacion/cliente-servidor.html>
- [25] «Client-server model». [En línea]. Disponible en: https://en.wikipedia.org/wiki/Client%E2%80%93server_model#Client_and_server_communication
- [26] «¿Qué patrón usa Angular? MVC o MVVM». [En línea]. Disponible en: <https://openwebinars.net/blog/que-patron-usa-angular-mvc-o-mvvm/>
- [27] «Documentación Django: FAQ». [En línea]. Disponible en: <https://docs.djangoproject.com/en/2.1/faq/general/#django-appears-to-be-a-mvc-framework-but-you-call-the-controller-the-view-and-the-view-the-template-how-come-you-don-t-use-the-standard-names>
- [28] «Material Design». [En línea]. Disponible en: <https://material.io/>
- [29] «Colores y tipografía UCM». [En línea]. Disponible en: <https://ssii.ucm.es/colores-y-tipografia>
- [30] «Chart.js». [En línea]. Disponible en: <https://www.chartjs.org/docs/latest/>
- [31] «Miga de pan». Disponible en: [https://es.wikipedia.org/wiki/Miga_de_pan_\(inform%C3%A1tica\)](https://es.wikipedia.org/wiki/Miga_de_pan_(inform%C3%A1tica))
- [32] «FullCalendar». [En línea]. Disponible en: <https://fullcalendar.io/>
- [33] «Moment.js». [En línea]. Disponible en: <http://momentjs.com/docs/>
- [34] «List of HTTP status codes». [En línea]. Disponible en: https://en.wikipedia.org/wiki/List_of_HTTP_status_codes
- [35] «Python Web Server Gateway Interface». [En línea]. Disponible en: <https://www.python.org/dev/peps/pep-0333/>
- [36] «Django REST Authentication». [En línea]. Disponible en: <https://www.django-rest-framework.org/api-guide/authentication/>
- [37] «How to use sessions». [En línea]. Disponible en: <https://docs.djangoproject.com/en/2.2/topics/http/sessions/>
- [38] «Django CORS headers». [En línea]. Disponible en: <https://pypi.org/project/django-cors-headers/>

[39] «Django Filter». [En línea]. Disponible en: <https://django-filter.readthedocs.io/en/master/>

[40] «Fixtures Django». [En línea]. Disponible en: <https://code.djangoproject.com/wiki/Fixtures>

[41] «JSON Web Token». [En línea]. Disponible en: <https://jwt.io/>

[42] «Heroku». [En línea]. Disponible en: <https://www.heroku.com/>

[43] «Trello». [En línea]. Disponible en: <https://trello.com/>

ANEXOS

Anexo I: Guía de uso

En este anexo se incluye una guía de uso en la que se explican cada una de las funcionalidades que se pueden realizar en la aplicación. Como existen dos tipos de usuarios, en primer lugar, se tratarán las que atañen a ambos, y después las particulares de un usuario administrador.

1. Iniciar sesión

Para poder acceder a las funcionalidades de la aplicación es necesario iniciar sesión. La sesión del usuario permanece activa durante un cierto tiempo. Después de caducar la sesión se requerirá al usuario que vuelva a introducir sus credenciales.

La pantalla de *login* (figura 122) se encuentra en la ruta “/login”, que se forzará en caso de que sea la primera vez que se haya abierto la aplicación, haya caducado la sesión o se haya cerrado manualmente. El usuario deberá introducir en los campos de texto sus credenciales para acceder y ser redirigido a la pantalla de *dashboard*. (Figura 123).

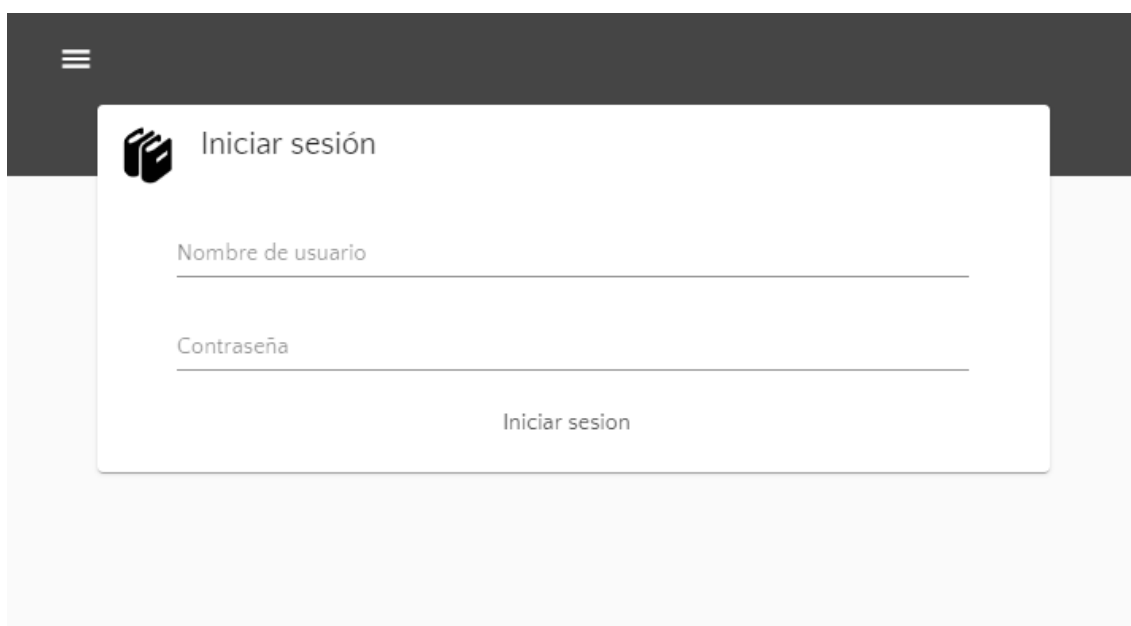


Figura 122: Pantalla de login

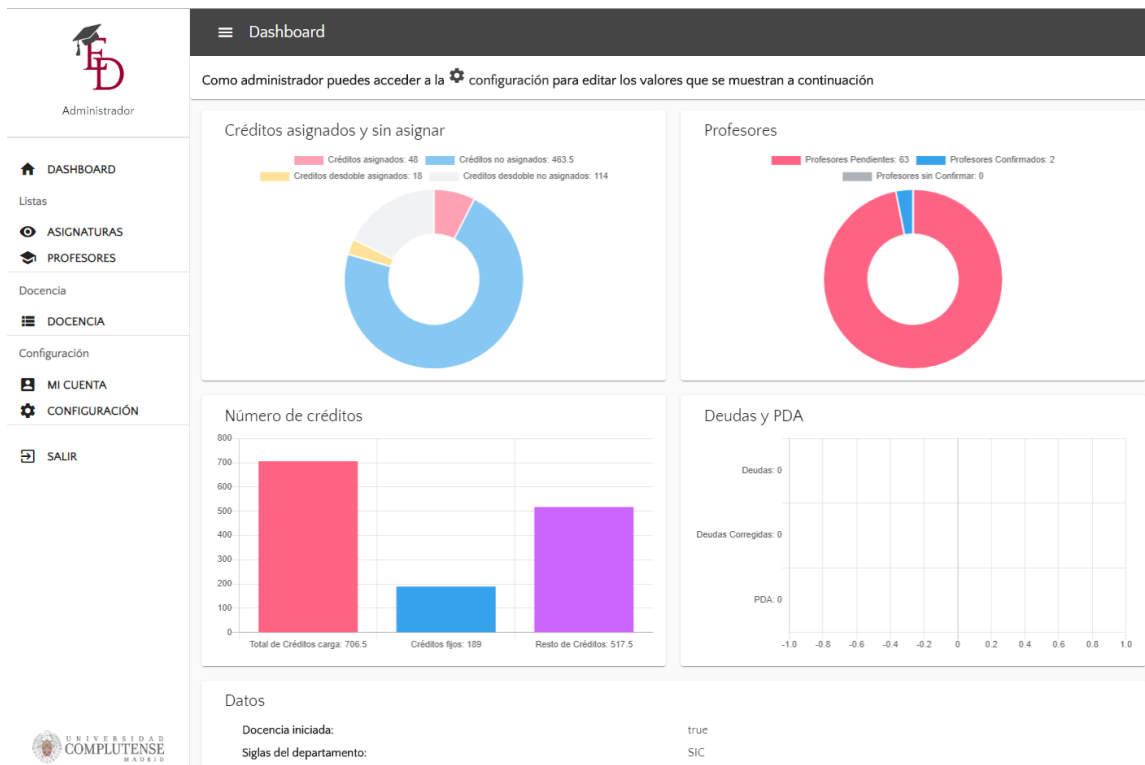


Figura 123: Pantalla de dashboard

2. Ver los detalles de una asignatura

Para ver los detalles de una asignatura es necesario acceder desde el menú lateral al apartado de asignaturas. La ruta de la página de asignaturas se encuentra en “/asignaturas” (figura 124).

Asignaturas

Buscar por nombre

Código	Siglas/Asignatura	Curso	Titulación
607300	[ABBD] Administración de bases de datos	2º	MII
803284	[ABD] Ampliación de bases de datos	3º	GI
803284	[ABD] Ampliación de bases de datos	3º	GI
803337	[AW] Aplicaciones web	4º	CS
607292	[ACFI] Auditoría, calidad y fiabilidad informáticas	1º	MII
803368	[NSQ] Bases de Datos noSQL	Opt.	GI/CC/GS
803273	[BD] Bases de datos	2º	GI
803273	[BD] Bases de datos	2º	GI/CC/GDV
803273	[BD] Bases de datos	2º	GI/CC/GDV
803273	[BD] Bases de datos	2º	GI/CC/GDV

+ Añadir nueva asignatura
 Importar asignaturas
 Exportar asignaturas

Mostrar

- Código
- Nombre
- Curso
- Grupo
- Cuatrimestre
- Titulación
- Departamento

Filtrar

Figura 124: Listado de asignaturas

Desde aquí solo se tiene que decidir una asignatura de la que se quieran ver los detalles y pulsar sobre ella para acceder a la vista de detalle que se puede ver en la figura 125.

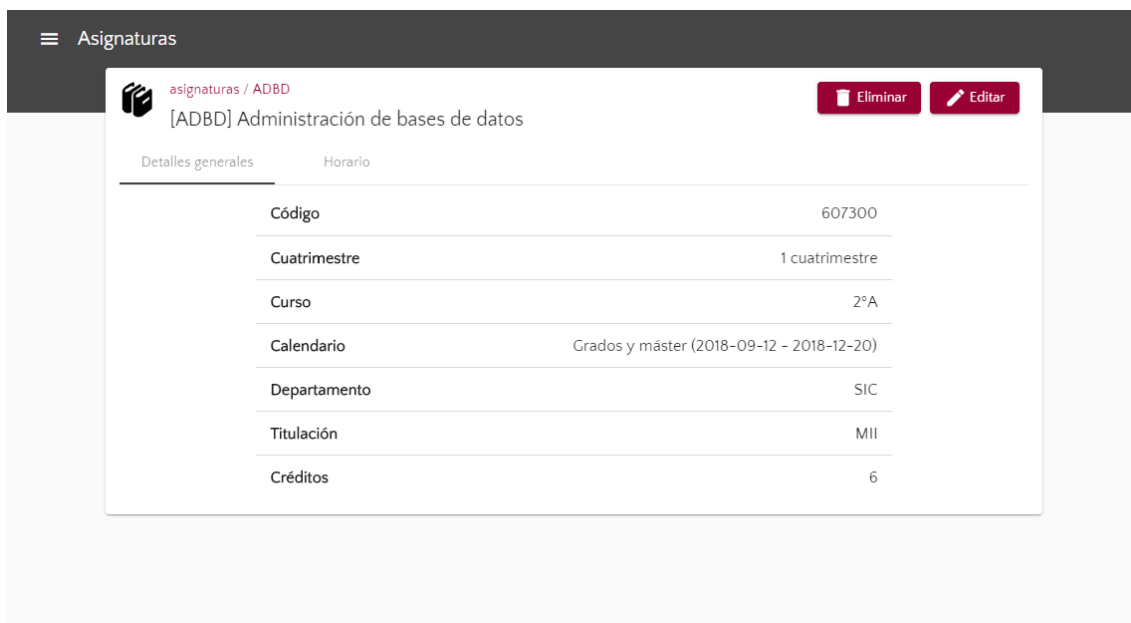


Figura 125: Página de detalle de una asignatura

Para ver el horario de la asignatura se ha de pulsar sobre la pestaña “horario”, que conducirá al usuario a una tabla como la que se puede ver en la figura 126.

	Lunes	Martes	Miercoles	Jueves	Viernes
09:00:00					
10:00:00					
11:00:00					
12:00:00					
13:00:00					
14:00:00					
15:00:00					
16:00:00					
17:00:00					
18:00:00		Aula 7		Aula 7	
19:00:00		Aula 7		Aula 7	
20:00:00					

Figura 126: Vista del horario de la asignatura desde el detalle

3. Buscar asignatura

Para buscar una asignatura hay que ubicarse en la lista de asignaturas, y desde las localizaciones marcadas que se pueden ver en la figura 127, introducir los parámetros de búsqueda. En la parte superior de la lista se puede buscar por nombre, y desde la parte derecha de la pantalla se pueden añadir filtros más concretos como el curso de la asignatura o las siglas.

Código	Siglas/Asignatura	Curso	Titulación
607300	[ADBD] Administración de bases de datos	2º	MII
803284	[ABD] Ampliación de bases de datos	3º	CI
803284	[ABD] Ampliación de bases de datos	3º	CI
803337	[AW] Aplicaciones web	4º	CS
607292	[ACFI] Auditoria, calidad y fiabilidad informáticas	1º	MII
803368	[NSQ] Bases de Datos noSQL	Cpt.	CI/CC/CS
803273	[BD] Bases de datos	2º	CI
803273	[BD] Bases de datos	2º	CI/CC/CDV
803273	[BD] Bases de datos	2º	CI/CC/CDV
803273	[BD] Bases de datos	2º	CI/CC/CDV
803333	[BD] Bases de datos	2º	CS/CDV
803225	[DA] Diseño de algoritmos	3º	CC
901558/803270	[EDA] Estructura de datos y algoritmos	2º	ADE-CI/CI/CC
901558/803270	[EDA] Estructura de datos y algoritmos	2º	ADE-CI/CI/CC
900213	[EDA] Estructura de datos y algoritmos	2º	DC

Figura 127: Buscar asignatura

4. Ver los detalles de un profesor

Para acceder a los detalles de un profesor, lo primero será pulsar sobre la opción “Profesores” del menú lateral. Seguidamente, al igual que con las asignaturas, se ha de pulsar sobre un profesor concreto. Cuando el profesor tenga asignaturas incluidas en su docencia, se mostrará un apartado con la elección que lleve hasta el momento. A continuación, se pueden ver en las figuras 128, 129 y 130 los pasos descritos.

Docencia	Escalafo	Apellidos	Nombre	Email	Categoría
✓	1	Frutos Escrig	David de	defrutos@sip.ucm. CU	
✓	2	Peña Mari	Ricardo	fake.ricardo@sip.u. CU	
	3	Martí Oliet	Narciso	fake.narciso@sip.u. CU	
	4	López Fraguas	Francisco J.	fake.fraguas@sip.u. CU	
	5	Núñez García	Manuel	fake.mn@sip.ucm. CU	
	6	Fernández Camacho	Maria Inés	fake.minesfc@sip.u. TU	
	7	Bradley Delso	Margarita	fake.bradley@sip.u. TU	
	8	Gil Luezas	Ana	fake.anagil@sip.uc TU	
	9	Ortega Mallén	Yolanda	fake.yolanda@sip.u. TU	

Figura 128: Listado de profesores



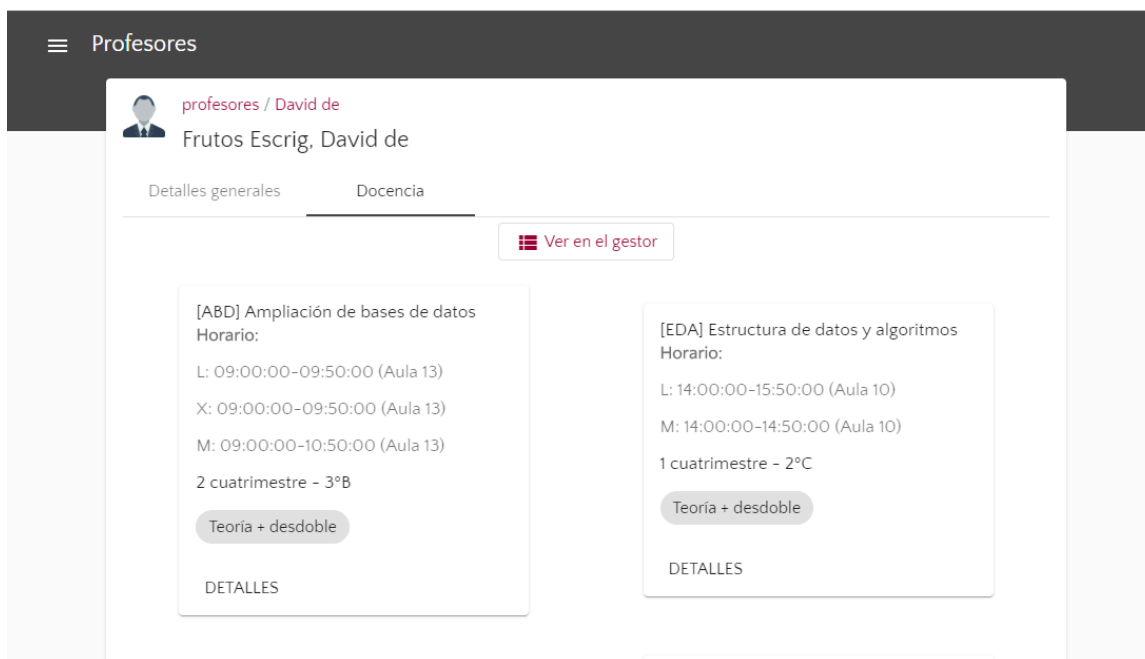
Profesores

profesores / David de
Frutos Escrig, David de

Detalles generales Docencia

Nombre	David de
Apellidos	Frutos Escrig
Email	defrutos@sip.ucm.es
Deuda	2014: 0 créditos 2015: 0 créditos 2016: 0 créditos 2017: 0 créditos
Teléfono	4529
Despacho	310C(Mat)
PDA	0 créditos
Categoría	Catedrático de Universidad
Departamento	SIC

Figura 129: Página de detalle de un profesor



Profesores

profesores / David de
Frutos Escrig, David de

Detalles generales Docencia

[Ver en el gestor](#)

[ABD] Ampliación de bases de datos
Horario:
L: 09:00:00-09:50:00 (Aula 13)
X: 09:00:00-09:50:00 (Aula 13)
M: 09:00:00-10:50:00 (Aula 13)
2 cuatrimestre - 3ºB
[Teoría + desdoble](#)
[DETALLES](#)

[EDA] Estructura de datos y algoritmos
Horario:
L: 14:00:00-15:50:00 (Aula 10)
M: 14:00:00-14:50:00 (Aula 10)
1 cuatrimestre - 2ºC
[Teoría + desdoble](#)
[DETALLES](#)

Figura 130: Vista de la docencia del profesor desde el detalle

5. Buscar profesor

Desde la parte superior izquierda del listado de profesores se puede realizar una búsqueda por nombre y apellidos. También se pueden aplicar filtros más concretos como los que hay en el listado de asignaturas. En la figura 131 se señalan las localizaciones desde donde se pueden realizar las búsquedas.

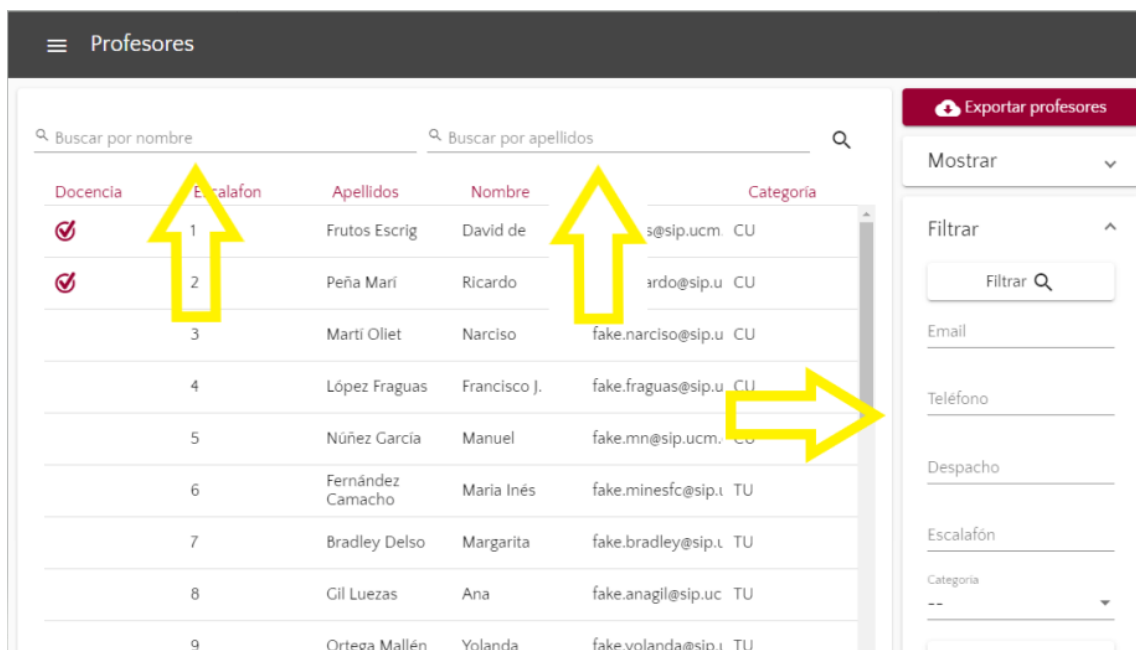


Figura 131: Buscar profesor

6. Cambiar contraseña

Para cambiar la contraseña, lo primero que hay que hacer es posicionarse en la pantalla de detalle del profesor que ha iniciado sesión. Desde ahí, pulsar sobre el botón “Cambiar contraseña” que aparece señalado en la figura 132. Una vez pulsado, solo se deberá introducir la contraseña actual y la nueva contraseña dos veces, y pulsar sobre el botón “Cambiar contraseña” (figura 133). Este botón solo estará habilitado cuando estén todos los campos rellenos y sin fallos.

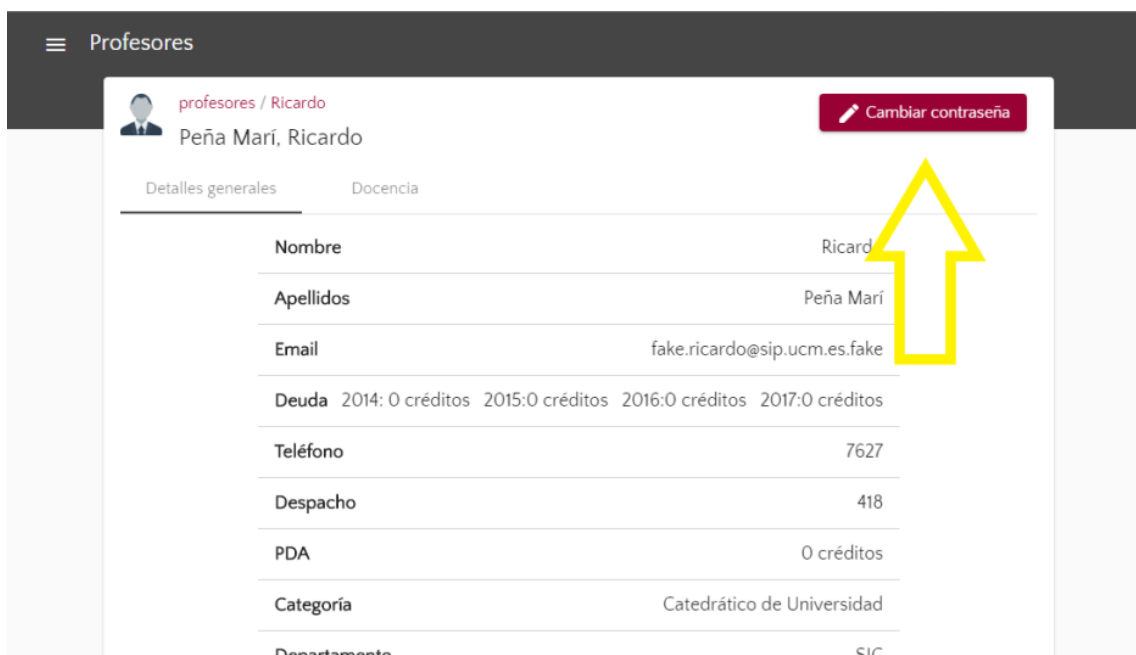


Figura 132: Botón para cambiar la contraseña

Profesores

← Cambiar contraseña

Contraseña actual *

Nueva contraseña *

Repetir nueva contraseña *

Cambiar contraseña

Figura 133: Formulario para cambiar la contraseña

7. Añadir asignatura a la elección

Cuando a un profesor le llega el turno de elegir la docencia del curso próximo, tiene que ir a la página dedicada a ello. Para llegar ahí hay dos opciones, desde el menú lateral en el apartado “Docencia” o desde la pantalla de *dashboard* en el enlace que se puede ver en el mensaje superior.

Una vez en la página objetivo, se ha de elegir las asignaturas pulsando sobre la *checkbox* o sobre el signo positivo si se trata de una asignatura divisible (ver figura 134).

Elección Docencia Filtrar asignaturas

Información

Esta elección está confirmada

Créditos seleccionados: 12 de 24.000 créditos

Deuda utilizada: 0 / Deuda disponible: 0

FDA utilizada: 0 / FDA disponible: 0

Asignaturas

- [ADBD] Administración de bases de datos
2018-09-12/2018-12-20
Martes: 18:00:00-19:50:00
Jueves: 18:00:00-19:50:00
Créditos disponibles: 6/6
- [ABD] Ampliación de bases de datos
2019-01-29/2019-05-17
Lunes: 09:00:00-09:50:00
Miércoles: 09:00:00-09:50:00
Martes: 09:00:00-10:50:00
Créditos: 6
- Divisible** [ABD] Ampliación de bases de datos
2019-01-29/2019-05-17
Martes: 09:00:00-10:50:00
Créditos: 3
- [ABD] Ampliación de bases de datos
2019-01-29/2019-05-17
Martes: 16:00:00-17:50:00
Miércoles: 16:00:00-17:50:00
Créditos: 6

Estado de la elección

1er Cuatri. 2º Cuatri. enero 2019

lun.	mar.	mié.	jue.	vie.
31	1	2	3	4
7	8	9	10	11
14	15	16	17	18
21	22	23	24	25

Figura 134: Añadir asignatura a la elección

8. Enviar docencia al administrador

Desde la página de la elección de docencia, si es el turno del profesor que ha iniciado sesión, entonces le aparecen tres botones en la parte inferior derecha de la pantalla (los cuales se muestran en la figura 135).

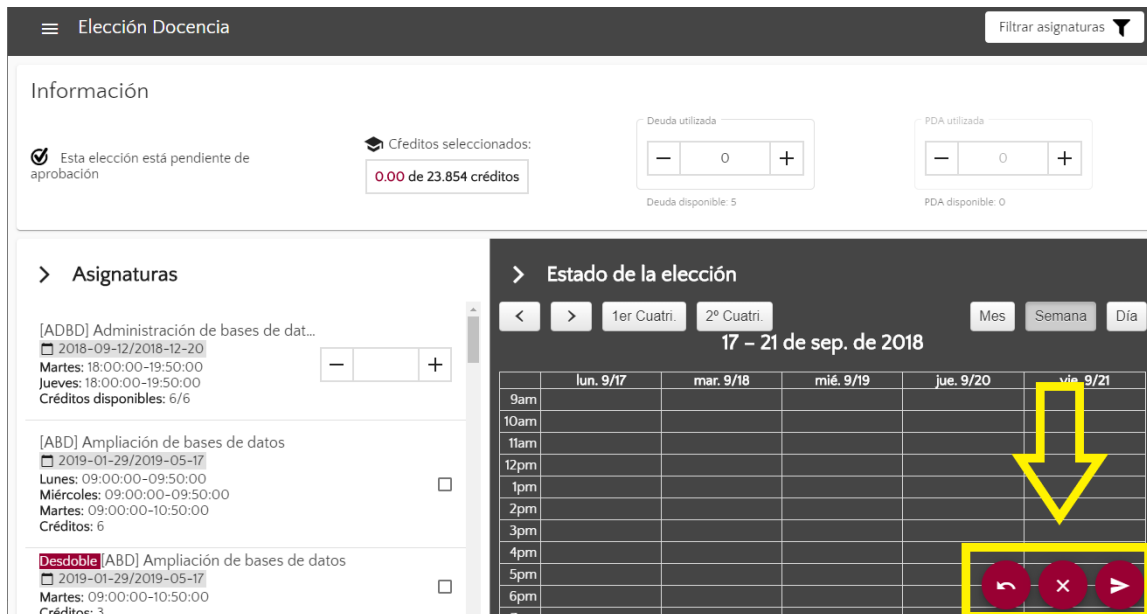


Figura 135: Botones elección docencia

El primero de los tres botones, al ser pulsado restaura la elección docente a la última versión que se haya guardado. El segundo limpia toda la elección. El tercero despliega una ventana emergente que permite al usuario escribir un mensaje al administrador (ver figura 136). Al pulsar en el botón “Enviar” se mandará la elección al administrador para que éste la pueda revisar y confirmar.

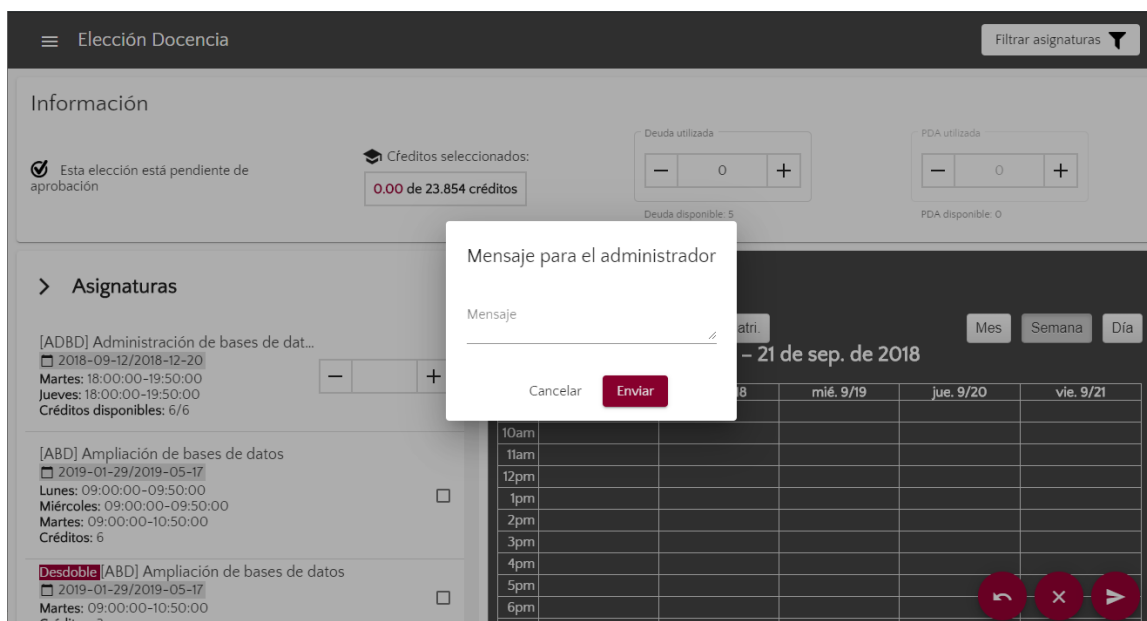
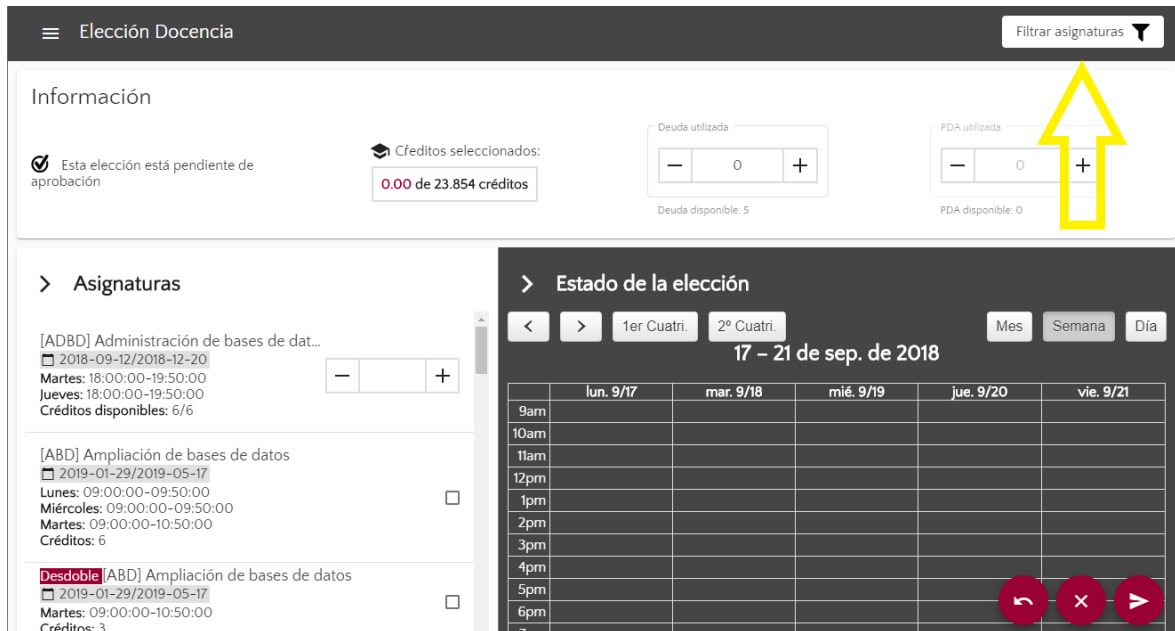


Figura 136: Mensaje para el administrador

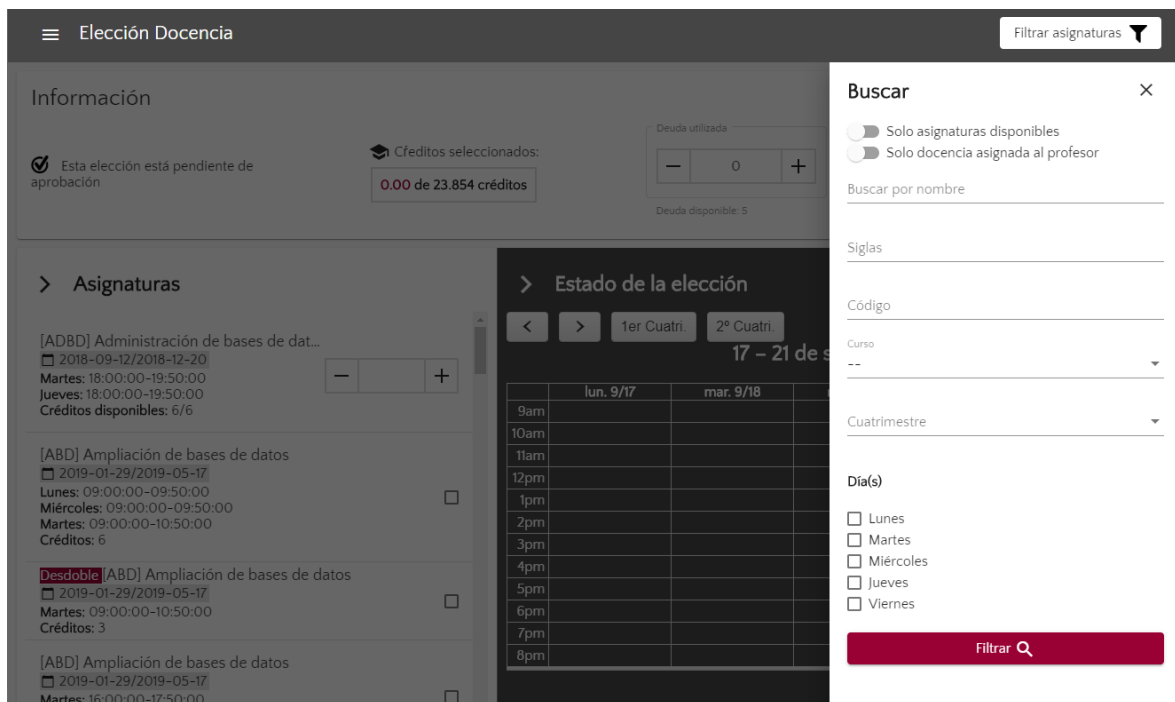
9. Buscar asignaturas desde la elección de docencia

Reutilizando el sistema de filtrado utilizado en el listado de asignaturas, la página de elección de docencia tiene filtros y formularios de búsqueda. Para acceder al menú de filtros hay que pulsar sobre el botón superior derecho que se indica en la figura 137. Una vez desplegado (ver figura 138) solo se deberá introducir el texto o las opciones que se deseen buscar.



The screenshot shows the 'Elección Docencia' interface. At the top right, there is a button labeled 'Filtrar asignaturas' with a downward arrow icon. A yellow arrow points to this button. Below the header, there is an 'Información' section with a status indicator 'Esta elección está pendiente de aprobación', a 'Códigos seleccionados' field showing '0.00 de 23.854 créditos', and two input fields for 'Deuda utilizada' and 'PDA utilizada', both showing '0'. Below this, there are two main sections: 'Asignaturas' on the left and 'Estado de la elección' on the right. The 'Estado de la elección' section shows a calendar for the period '17 - 21 de sep. de 2018' with columns for 'lun. 9/17', 'mar. 9/18', 'mié. 9/19', 'jue. 9/20', and 'vie. 9/21'. The 'Asignaturas' section lists several courses with their respective schedules and credit availability.

Figura 137: Botón filtrar asignaturas desde la elección de docencia

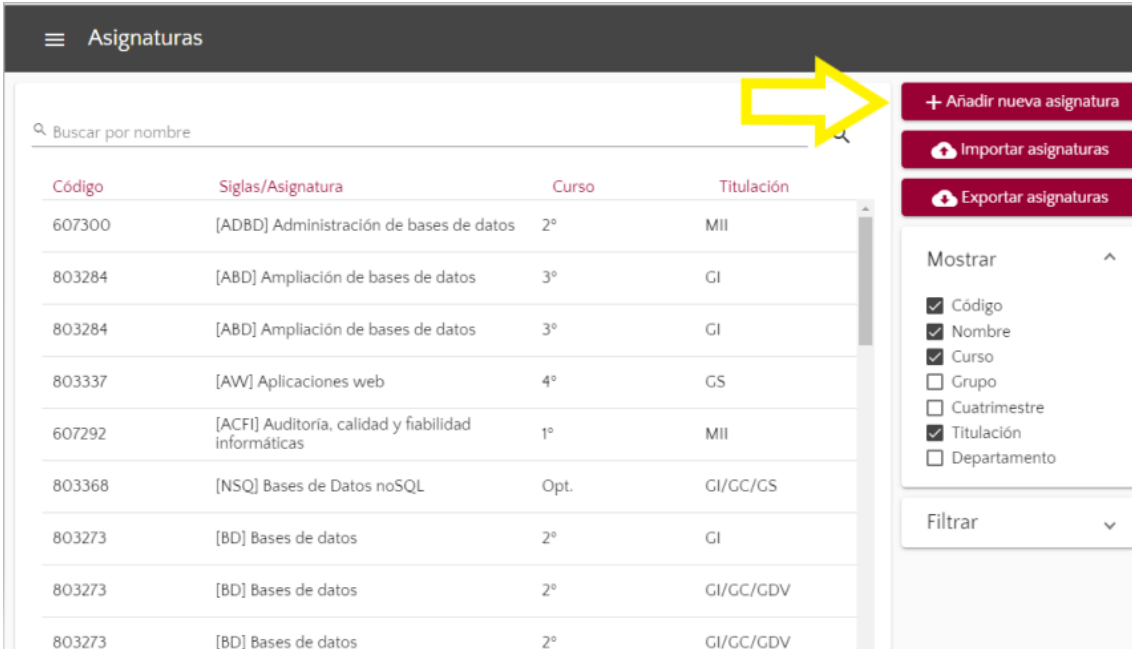


The screenshot shows the 'Elección Docencia' interface with the 'Filtrar asignaturas' menu open. The menu is titled 'Buscar' and contains several filter options: 'Solo asignaturas disponibles' (checked), 'Solo docencia asignada al profesor' (unchecked), 'Buscar por nombre' (input field), 'Siglas' (input field), 'Código' (input field), 'Curso' (dropdown menu), 'Cuatrimestre' (dropdown menu), and 'Día(s)' (checkboxes for 'Lunes', 'Martes', 'Miércoles', 'Jueves', and 'Viernes'). A red 'Filtrar' button is at the bottom of the menu. The background interface is dimmed, showing the same 'Información' and 'Asignaturas' sections as in Figure 137.

Figura 138: Filtros de las asignaturas desde la elección de docencia

10. Administrador - Añadir/Editar asignatura

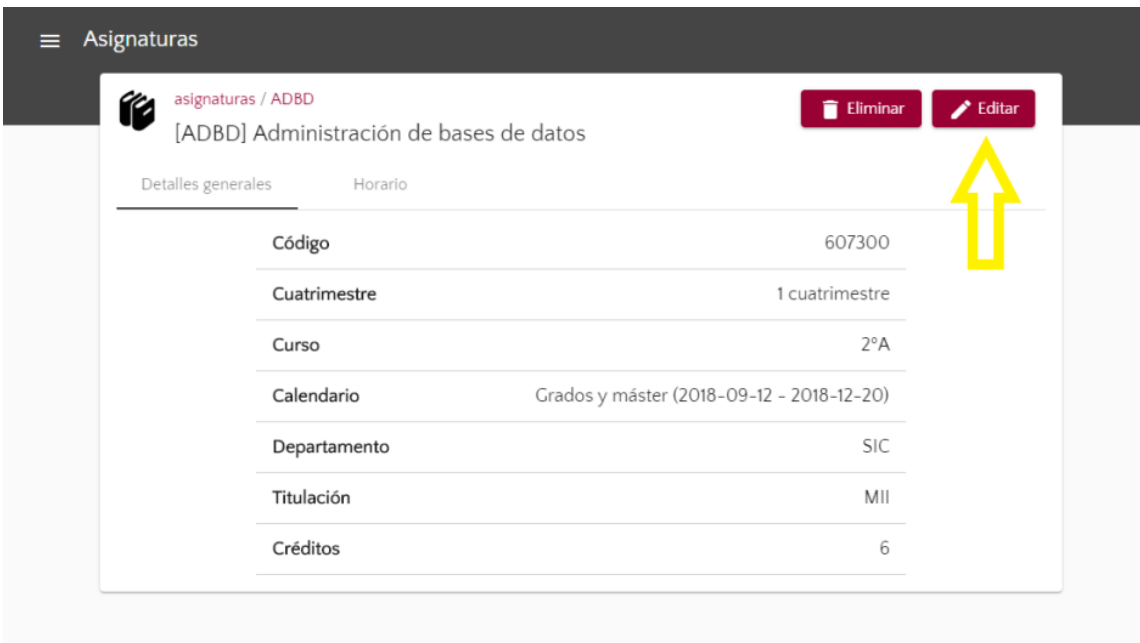
Para añadir o editar una asignatura, el formulario utilizado es el mismo, así que se explicarán ambas pantallas en este mismo apartado. Para añadir una asignatura hay que ubicarse en el listado, y después hay que pulsar sobre el botón superior derecho “Añadir nueva asignatura” (ver figura 139).



The screenshot shows the 'Asignaturas' management interface. At the top, there is a search bar labeled 'Buscar por nombre'. Below it is a table with columns: Código, Siglas/Asignatura, Curso, and Titulación. The table lists several courses, including '607300 [ABDD] Administración de bases de datos' and '803284 [ABD] Ampliación de bases de datos'. To the right of the table is a sidebar with three main buttons: '+ Añadir nueva asignatura', 'Importar asignaturas', and 'Exportar asignaturas'. Below these buttons is a 'Mostrar' section with checkboxes for 'Código', 'Nombre', 'Curso', 'Grupo', 'Cuatrimestre', 'Titulación', and 'Departamento'. A 'Filtrar' dropdown is also present. A yellow arrow points to the '+ Añadir nueva asignatura' button.

Figura 139: Botón de añadir asignatura

Para editar una asignatura hay que acceder a una asignatura y desde ahí pulsar sobre el botón editar que se muestra en la figura 140.

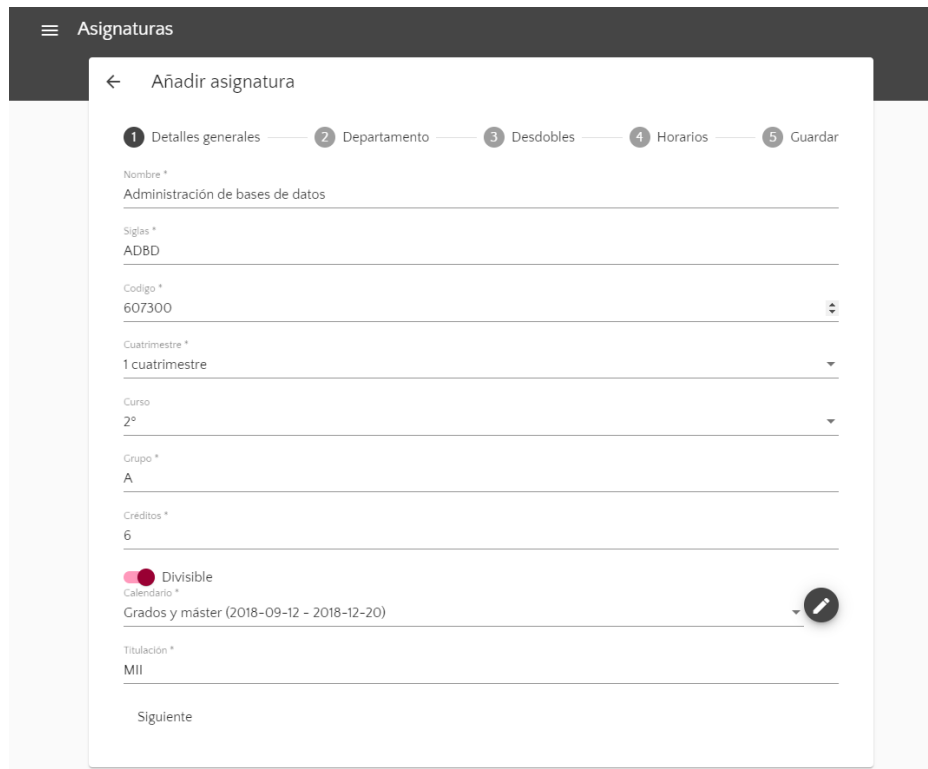


The screenshot shows the 'Asignaturas' management interface with a modal window open for editing a course. The modal title is 'asignaturas / ADBD' and the course name is '[ADBD] Administración de bases de datos'. At the top right of the modal are two buttons: 'Eliminar' and 'Editar'. A yellow arrow points to the 'Editar' button. Below the buttons is a 'Detalles generales' section with the following details:

Código	607300
Cuatrimestre	1 cuatrimestre
Curso	2ºA
Calendario	Grados y máster (2018-09-12 - 2018-12-20)
Departamento	SIC
Titulación	MII
Créditos	6

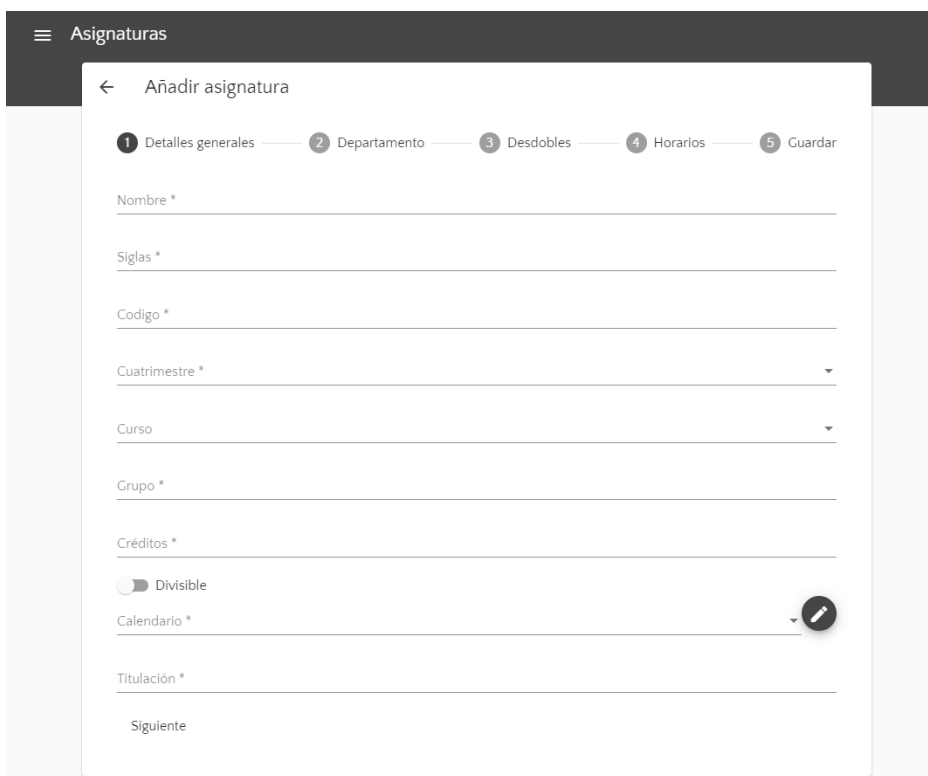
Figura 140: Botón editar asignatura

Una vez en el formulario, ya sea editando (figura 141) o añadiendo una nueva asignatura (figura 142), se han de completar todos los campos obligatorios y pulsar el botón que salva los cambios como se puede observar en la figura 143.



The screenshot shows a mobile application interface for adding a subject. At the top, there is a dark header with a hamburger menu icon and the text 'Asignaturas'. Below this is a white modal window titled 'Añadir asignatura' with a back arrow on the left. A progress indicator at the top of the modal shows five steps: 1. Detalles generales (highlighted), 2. Departamento, 3. Desdobles, 4. Horarios, and 5. Guardar. The form contains the following fields: 'Nombre *' with the value 'Administración de bases de datos'; 'Siglas *' with the value 'ADBD'; 'Codigo *' with the value '607300' and a dropdown arrow; 'Cuatrimestre *' with the value '1 cuatrimestre' and a dropdown arrow; 'Curso' with the value '2º' and a dropdown arrow; 'Grupo *' with the value 'A'; 'Créditos *' with the value '6'; a 'Divisible' toggle switch which is turned on; 'Calendario *' with the value 'Grados y máster (2018-09-12 - 2018-12-20)' and a dropdown arrow; and 'Titulación *' with the value 'MII'. At the bottom of the modal is a 'Siguiente' button. A pencil icon in a circle is visible on the right side of the 'Calendario' field, indicating edit mode.

Figura 141: Editar asignatura



The screenshot shows the same mobile application interface for adding a subject, but in add mode. The header and modal title are the same. The progress indicator shows the same five steps. The form fields are empty: 'Nombre *', 'Siglas *', 'Codigo *', 'Cuatrimestre *', 'Curso', 'Grupo *', 'Créditos *', 'Divisible' (toggle switch is turned off), 'Calendario *', and 'Titulación *'. The 'Siguiente' button is at the bottom. The pencil icon in a circle is visible on the right side of the 'Calendario' field, indicating add mode.

Figura 142: Añadir asignatura

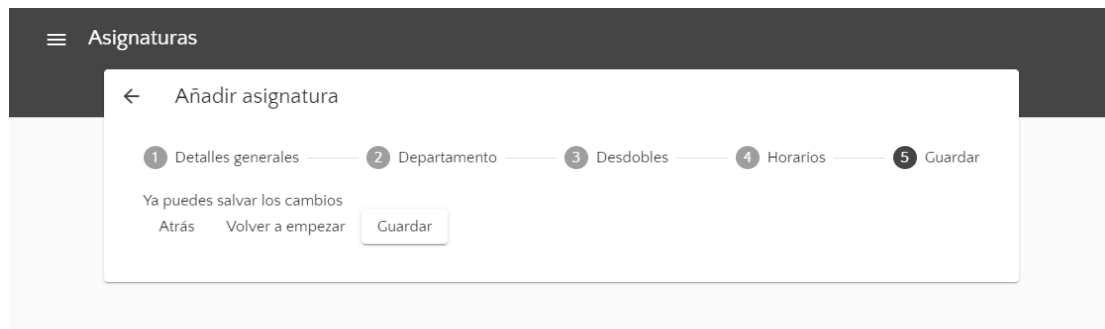


Figura 143: Botón guardar edición o nueva asignatura

11. Administrador - Añadir/Editar profesor

Al igual que en el apartado anterior, para añadir o editar un profesor solo hay un formulario, con la diferencia de que uno está vacío y el otro está relleno de datos. Para añadir un profesor hay que posicionarse en la página del listado de profesores. A continuación, hay que pulsar sobre el botón superior derecho “Añadir nuevo profesor” (ver figura 144).

Profesores

Buscar por nombre Buscar por apellidos

+ Añadir nuevo profesor

Importar profesores

Exportar profesores

Mostrar

- Docencia
- Escalafon
- Nombre
- Email
- Teléfono
- Despacho
- Departamento
- Categoría

Filtrar

Docencia	Escalafon	Apellidos	Nombre	Email	Categoría
	1	Frutos Escrig	David de	defrutos@sip.ucm. CU	
	2	Peña Mari	Ricardo	fake.ricardo@sip.u CU	
	3	Martí Olliet	Narciso	fake.narciso@sip.u CU	
	4	López Fraguas	Francisco J.	fake.fraguas@sip.u CU	
	5	Núñez García	Manuel	fake.mn@sip.ucm. CU	
	6	Fernández Camacho	María Inés	fake.minesfc@sip.i TU	
	7	Bradley Delso	Margarita	fake.bradley@sip.i TU	
	8	Gil Luezas	Ana	fake.anagil@sip.uc TU	
	9	Ortega Mallén	Yolanda	fake.yolanda@sip.i TU	

Figura 144: Botón para añadir profesor

Si se desea editar un profesor, habrá que ir a la vista de detalle y pulsar sobre el botón de editar que aparece señalado en la figura 145.

profesores / David de

Frutos Escrig, David de

Eliminar Editar

Detalles generales Docencia

Nombre	David de
Apellidos	Frutos Escrig
Email	defrutos@sip.ucm.es
Deuda	2014: 0 créditos 2015: 0 créditos 2016: 0 créditos 2017: 0 créditos
Teléfono	4529
Despacho	310C(Mat)
PDA	0 créditos
Categoría	Catedrático de Universidad
Departamento	SIC

Figura 145: Botón editar profesor

Una vez en el formulario, ya sea editando (figura 146) o añadiendo un nuevo profesor (figura 147) se han de completar todos los campos obligatorios y pulsar el botón de guardar al final (ver figura 148).

Profesores

← Editar profesor

1 Detalles generales 2 Deuda & PDA 3 Guardar

Administrador

Nombre *
David de

Apellidos *
Frutos Escrig

Despacho *
310C(Mat)

Email *
defrutos@sip.ucm.es

Teléfono *
4529

Escalafón *
1

Categoría

Figura 146: Editar profesor

Profesores

← Añadir profesor

1 Detalles generales 2 Deuda & PDA 3 Guardar

Administrador

Nombre *

Apellidos *

Despacho *

Email *

Teléfono *

Escalafón *

Figura 147: Añadir profesor

Profesores

← Añadir profesor

1 Detalles generales 2 Deuda & PDA 3 Guardar

Ya puedes salvar los cambios

Atrás Volver a empezar Guardar

Figura 148: Botón guardar edición o nuevo profesor

12. Administrador - Confirmar o eliminar docencia

Cuando el profesor le envía la docencia al administrador para que se la confirme, éste tiene que acceder a la página de docencia. Desde ahí, ha de seleccionar al profesor (figura 149) que desea validar. En la parte inferior derecha de la pantalla aparecen cuatro botones: el primero elimina la docencia del profesor definitivamente tras un mensaje de confirmación, el segundo restaura la docencia a la última que se haya guardado, el tercero limpia esa docencia para restablecer los valores por defecto, y por último el cuarto permite confirmar la docencia (figura 149). A partir de este momento pasará el turno al siguiente profesor.

Elección Docencia Filtrar asignaturas

Elección de docencia de: David de Frutos Esc...

Información

Esta elección está pendiente de aprobación

Céditos seleccionados: **24.00 de 24.000 créditos**

Deuda utilizada: Deuda disponible: 0

PDA utilizada: PDA disponible: 0

Asignaturas

- [MDL] Matemática Discreta y Lógica Matemática
2018-09-12/2018-12-20
Lunes: 09:00:00-10:50:00
Martes: 09:00:00-10:50:00
Créditos: 6
- [MDL] Matemática Discreta y Lógica Matemática
2019-01-29/2019-05-17
Lunes: 09:00:00-10:50:00
Miércoles: 09:00:00-10:50:00
Créditos: 6
- [MDL] Matemática Discreta y Lógica Matemática
2018-09-12/2018-12-20
Lunes: 11:30:00-12:20:00
Martes: 11:30:00-12:20:00
Jueves: 11:30:00-12:20:00

Estado de la elección

1er Cuatri. 2º Cuatri. Mes Semana Día

28 de ene. - 1 de feb. de 2019

	lun. 1/28	mar. 1/29	mié. 1/30	jue. 1/31	vie. 2/1
9am					
10am			9 - 10:50 [MDL] Matemática Discreta y Lógica		
11am	11:30 - [MDL] Matemática	11:30 - [MDL] Matemática		11:30 - [MDL] Matemática	12 - [MDL] Matemática
12pm					
1pm					
2pm					
3pm					
4pm					
5pm					
6pm					
7pm					

Figura 149: Confirmar la docencia

13. Administrador - Cambiar configuración

Para cambiar la configuración del departamento hay que dirigirse al menú lateral y pulsar sobre el apartado de Configuración. Se abrirá una página con un formulario con la configuración existente (ver figura 150). Solamente se tienen que editar los campos deseados y pulsar el botón “Guardar” situado en la parte inferior de esta pantalla.

Configuration

Departamento

*Ojo! Algún cambio puede desconfigurar el sistema, utilizar con precaución

Siglas *

SIC

Nombre del departamento *

Sistemas Informáticos y Computación

Total de créditos *

706.5

Deudas *

0

Deudas Corregidas *

0

PDA *

0

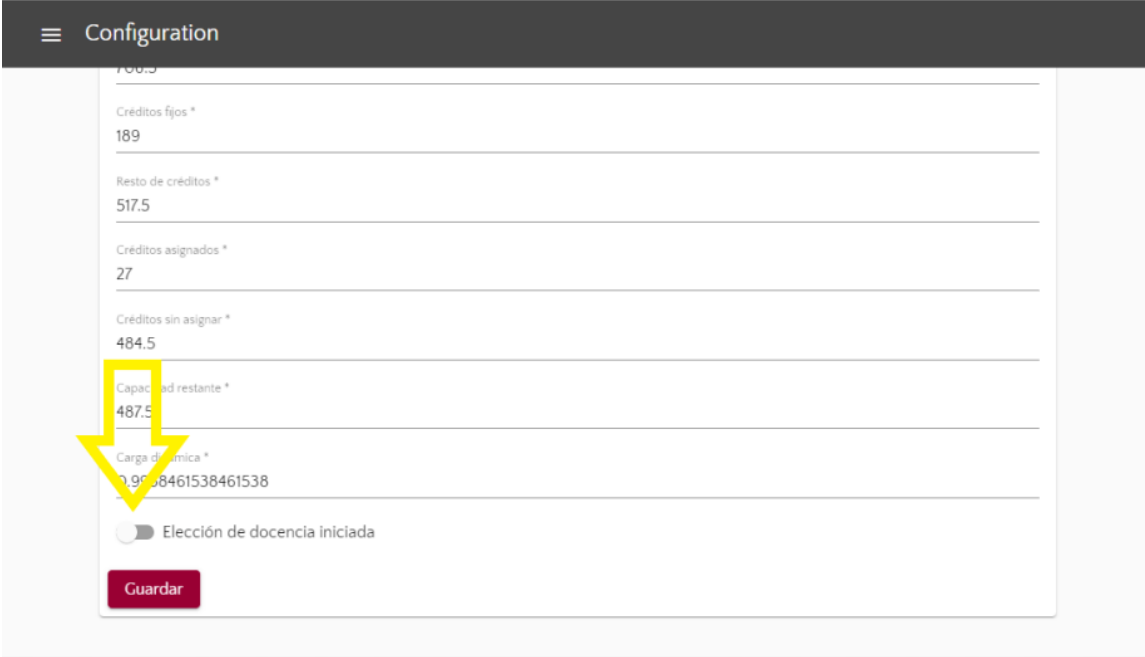
Total créditos carga *

706.5

Figura 150: Cambiar configuración del departamento

14. Administrador - Iniciar y reiniciar docencia

Desde la pantalla de configuración, previamente mencionada, una de las opciones disponibles es la de iniciar el proceso de elección de docencia. Para ello solo hay que pulsar sobre el interruptor y guardar los cambios (ver figura 151). Para reiniciar la docencia hay que pulsar sobre el botón señalado en la figura 152 y en la ventana emergente pulsar sobre “Reiniciar” (figura 153).




The screenshot shows a 'Configuration' screen with a dark header. Below the header, there is a list of fields with their respective values:

- Créditos fijos *: 189
- Resto de créditos *: 517.5
- Créditos asignados *: 27
- Créditos sin asignar *: 484.5
- Capacidad restante *: 487.5
- Carga docente *: 0.9958461538461538

At the bottom of the list, there is a toggle switch labeled 'Elección de docencia iniciada' which is currently turned off. Below the toggle is a red button labeled 'Guardar'. A yellow arrow points to the 'Capacidad restante' field.

Figura 151: Iniciar elección de docencia



The screenshot shows a 'Configuration' screen with a dark header. Below the header, there is a form for department configuration. The form includes the following fields and values:

- Departamento: *Ojo! Algún cambio puede desconfigurar el sistema, utilizar con precaución
- Siglas *: SIC
- Nombre del departamento *: Sistemas Informáticos y Computación
- Total de créditos *: 706.5
- Deudas *: 0
- Deudas Corregidas *: 0
- PDA *: 0
- Total créditos carga *: 706.5

At the top right of the form, there is a red button labeled 'Reiniciar docencia'. A yellow arrow points to this button.

Figura 152: Botón para reiniciar la elección de docencia

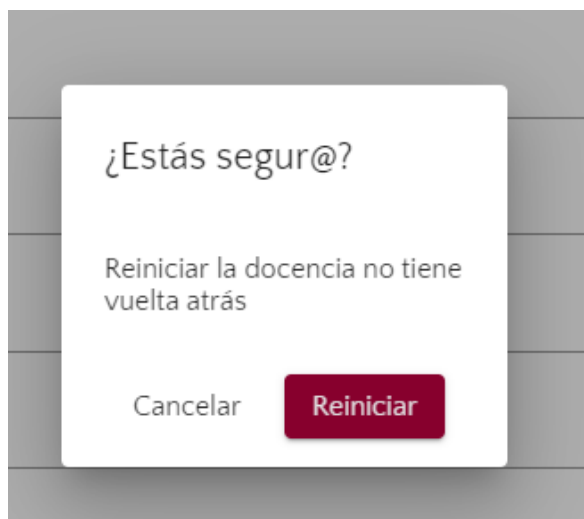


Figura 153: Popup para reiniciar la elección de docencia

15. Administrador - Importar y exportar lista de asignaturas

Para exportar el listado de asignaturas en formato “.xlsx” habrá que ubicarse en la página del listado de asignaturas. Desde ahí se ha de pulsar en el botón exportar y esperar a que la descarga comience (ver figura 154).

Código	Siglas/Asignatura	Curso	Titulación
607300	[ABBD] Administración de bases de datos	2º	MII
803284	[ABD] Ampliación de bases de datos	3º	GI
803284	[ABD] Ampliación de bases de datos	3º	GI
803337	[AW] Aplicaciones web	4º	GS
607292	[ACFI] Auditoría, calidad y fiabilidad informáticas	1º	MII
803368	[NSQ] Bases de Datos noSQL	Opt.	GI/GC/GS
803273	[BD] Bases de datos	2º	GI
803273	[BD] Bases de datos	2º	GI/GC/GDV
803273	[BD] Bases de datos	2º	GI/GC/GDV

Figura 154: Botón para exportar asignaturas

Para importar una o varias asignaturas el proceso es similar. En la parte superior derecha de la lista de asignaturas, hay que pulsar sobre el botón “Importar asignaturas” que aparece señalado en la figura 155. A continuación, aparecerá una página en la que se deberá completar y adjuntar el archivo en formato “.xlsx” (Figura 156).

Asignaturas

Buscar por nombre

Código	Siglas/Asignatura	Curso	Titulación
607300	[ADB] Administración de bases de datos	2º	MII
803284	[ABD] Ampliación de bases de datos	3º	GI
803284	[ABD] Ampliación de bases de datos	3º	GI
803337	[AW] Aplicaciones web	4º	GS
607292	[ACFI] Auditoría, calidad y fiabilidad informáticas	1º	MII
803368	[NSQ] Bases de Datos noSQL	Opt.	GI/GC/GS
803273	[BD] Bases de datos	2º	GI
803273	[BD] Bases de datos	2º	GI/GC/GDV
803273	[BD] Bases de datos	2º	GI/GC/GDV

+ Añadir nueva asignatura

Importar asignaturas

Exportar asignaturas

Mostrar

- Código
- Nombre
- Curso
- Grupo
- Cuatrimestre
- Titulación
- Departamento

Filtrar

Figura 155: Botón para importar asignaturas

Asignaturas

Importar asignaturas

Puedes importar el listado de asignaturas y desdobles desde una hoja de cálculo de Excel 2010 (.xlsx).

Tienes un ejemplo de la estructura que debe de tener el fichero para su correcta importación en [este enlace](#).

Siglas del departamento
Ejemplo: SIC

Nombre del departamento
Ejemplo: Sistemas Informáticos y Computación

Año académico
Ejemplo: 2018 (para el año académico 2018-2019)

Sobrescribir asignaturas existentes

Selecciona un archivo Importar

Figura 156: Página para importar asignaturas

16. Administrador - Importar y exportar lista de profesores

Para exportar el listado de profesores en formato “.xlsx” habrá que ubicarse en la página del listado de profesores. Desde ahí se ha de pulsar en el botón exportar y esperar a que la descarga comience (ver figura 157).

Profesores

Buscar por nombre Buscar por apellidos

Docencia	Escalafon	Apellidos	Nombre	Email	Categoria
<input checked="" type="checkbox"/>	1	Frutos Escrig	David de	defrutos@sip.ucm	CU
	2	Peña Marí	Ricardo	fake.ricardo@sip.u	CU
	3	Martí Oliet	Narciso	fake.narciso@sip.u	CU
	4	López Fraguas	Francisco J.	fake.fraguas@sip.u	CU
	5	Núñez García	Manuel	fake.mn@sip.ucm.	CU
	6	Fernández Camacho	María Inés	fake.minesfc@sip.i	TU
	7	Bradley Delso	Margarita	fake.bradley@sip.i	TU
	8	Gil Luezas	Ana	fake.anagil@sip.uc	TU
	9	Ortega Mallén	Yolanda	fake.yolanda@sip.i	TU

+ Añadir nuevo profesor
 Importar profesores
 Exportar profesores

Mostrar

- Docencia
- Escalafon
- Nombre
- Email
- Teléfono
- Despacho
- Departamento
- Categoría

Filtrar

Figura 157: Botón para exportar profesores

Para importar uno o varios profesores el proceso es similar. En la parte superior derecha hay que pulsar sobre el botón “Importar profesores” que aparece señalado en la figura 158. A continuación, aparecerá una página en la que se deberá completar y adjuntar el archivo en formato “.xlsx” (Figura 159).

Profesores

Buscar por nombre Buscar por apellidos

Docencia	Escalafon	Apellidos	Nombre	Email	Categoria
<input checked="" type="checkbox"/>	1	Frutos Escrig	David de	defrutos@sip.ucm	CU
	2	Peña Marí	Ricardo	fake.ricardo@sip.u	CU
	3	Martí Oliet	Narciso	fake.narciso@sip.u	CU
	4	López Fraguas	Francisco J.	fake.fraguas@sip.u	CU
	5	Núñez García	Manuel	fake.mn@sip.ucm.	CU
	6	Fernández Camacho	María Inés	fake.minesfc@sip.i	TU
	7	Bradley Delso	Margarita	fake.bradley@sip.i	TU
	8	Gil Luezas	Ana	fake.anagil@sip.uc	TU
	9	Ortega Mallén	Yolanda	fake.yolanda@sip.i	TU

+ Añadir nuevo profesor
 Importar profesores
 Exportar profesores

Mostrar

- Docencia
- Escalafon
- Nombre
- Email
- Teléfono
- Despacho
- Departamento
- Categoría

Filtrar

Figura 158: Botón para importar profesores

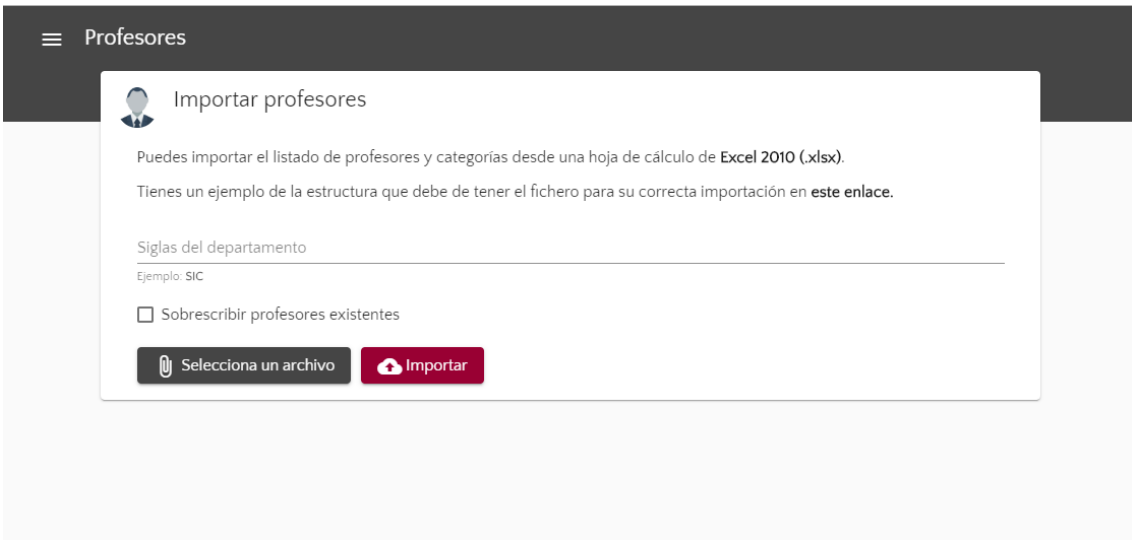


Figura 159: Página para importar profesores

Anexo II: Guía de instalación de la aplicación

1. Instalación de la aplicación web

Para el despliegue de la aplicación se va a utilizar un servidor Apache en un sistema Linux.

Antes de comenzar, los requisitos necesarios para la compilación del código de fuente son tener instalados Node.js en su versión 8.x o 10.x y el gestor de paquetes npm. Para comprobar la versión de Node.js instalada se puede ejecutar el siguiente comando en un terminal:

```
$ node -v
```

El gestor de paquetes npm viene incluido por defecto con Node.js. Para comprobarlo:

```
$ npm -v
```

El siguiente paso es acceder a la carpeta que contiene el proyecto Angular de la aplicación web e instalar todas las dependencias necesarias:

```
$ cd tfg-angular
```

```
$ npm install
```

Posteriormente, se compila el código y se copia al directorio que esté sirviendo el servidor web. Los resultados de la compilación del código se encuentran por defecto en la carpeta `/dist`.

```
$ ng build --prod
```

Finalmente, es necesario crear un archivo `.htaccess` en el directorio raíz que sirve el servidor web. Debido a las particularidades de Angular hay que redirigir todas las peticiones “fallidas” a `index.html`.

```
RewriteEngine On
```

```
# If an existing asset or directory is requested go to  
it as it is
```

```
RewriteCond %{DOCUMENT_ROOT}%{REQUEST_URI} -f [OR]
```

```
RewriteCond %{DOCUMENT_ROOT}%{REQUEST_URI} -d
```

```
RewriteRule ^ - [L]
```

```
# If the requested resource doesn't exist, use index.html
```

```
RewriteRule ^ /index.html
```

2. Instalación de la API

Para la instalación de la API se va a utilizar una máquina con el sistema operativo Debian 9. Sin embargo, esta guía sirve para cualquier servidor que conste de un sistema operativo basado en la arquitectura de Debian o similares.

Los requisitos fundamentales para desplegar la API son tener el intérprete de Python³, pip (administrador de paquetes), virtualenv y Apache2 instalados en el sistema. El intérprete de Python suele venir incorporado en las distribuciones de Linux más actuales.

El primer paso es comprobar que el sistema cuenta con las últimas actualizaciones:

```
sudo apt-get update -y
sudo apt-get upgrade -y
```

Después, es necesario revisar que el sistema cuenta con la versión 3 del intérprete de Python:

```
python3 --version
```

A continuación, se instala la herramienta que permitirá crear entornos virtuales de Python, de manera que será posible aislar las dependencias de la aplicación de otras aplicaciones que puedan estar siendo desarrolladas o servidas desde la misma máquina física.

```
sudo pip3 install virtualenv
virtualenv --python python3 DocenciaEnv
```

Se arranca el entorno virtual:

```
source DocenciaEnv/bin/activate
```

Se procede a instalar todas las dependencias necesarias para el correcto funcionamiento de la API:

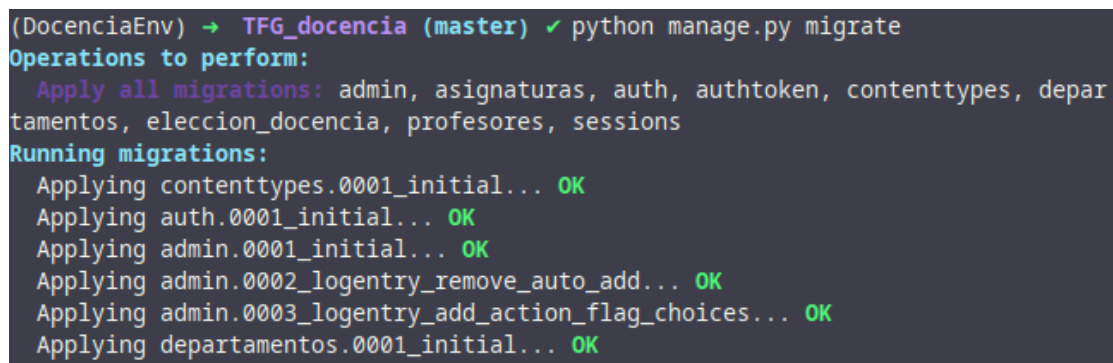
```
pip install Django
pip install djangorestframework
pip install django-cors-headers
pip install django-filter
pip install django-sendgrid-v5
pip install djangorestframework-simplejwt
pip install drf-nested-routers
pip install openpyxl
```

³ La versión utilizada en el servidor de referencia es la de Python 3.5.3

```
pip install coreapi
pip install coreschema
```

A continuación, se ejecutan los comandos que crearán la estructura de la base de datos (ver figura 160):

```
cd TFG_docencia
python manage.py makemigrations
python manage.py migrate
python manage.py loaddata departamentos_fixture.json
```



```
(DocenciaEnv) → TFG_docencia (master) ✓ python manage.py migrate
Operations to perform:
  Apply all migrations: admin, asignaturas, auth, authtoken, contenttypes, departamentos, eleccion_docencia, profesores, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying departamentos.0001_initial... OK
```

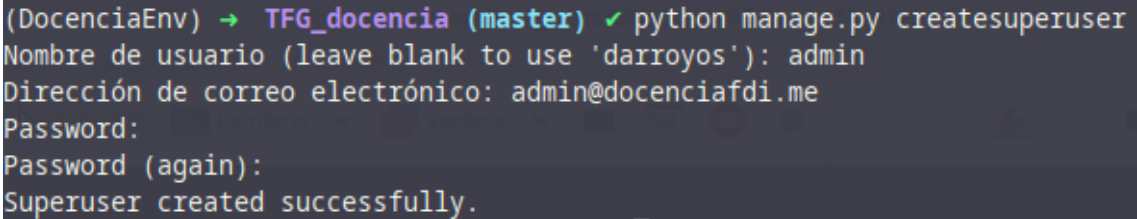
Figura 160: Creación de la estructura de la base de datos

Al realizar este paso, es necesario dar ciertos permisos de escritura sobre algunos directorios del proyecto para que el servidor web opere correctamente. Para ello, se incluye un script junto con el código de fuente que facilita la tarea, aunque también se pueden ejecutar manualmente los siguientes comandos:

```
chmod g+w TFG_docencia
sudo chown :www-data TFG_docencia
chmod g+w TFG_docencia/db.sqlite3
sudo chown :www-data TFG_docencia/db.sqlite3
chmod g+w -R TFG_docencia/uploads
sudo chown -R :www-data TFG_docencia/uploads
chmod g+w -R TFG_docencia/asignaturas/fixtures
sudo chown -R :www-data TFG_docencia/asignaturas/fixtures
chmod g+w -R TFG_docencia/departamentos/fixtures
sudo chown -R :www-data TFG_docencia/departamentos/fixtures
chmod g+w -R TFG_docencia/profesores/fixtures
sudo chown -R :www-data TFG_docencia/profesores/fixtures
```

Posteriormente, se debe crear un *superusuario* en la aplicación que pueda autenticarse en la API y realizar la importación de profesores y de asignaturas (ver figura 161):

```
python manage.py createsuperuser
```



```
(DocenciaEnv) → TFG_docencia (master) ✓ python manage.py createsuperuser
Nombre de usuario (leave blank to use 'darroyos'): admin
Dirección de correo electrónico: admin@docenciafdi.me
Password:
Password (again):
Superuser created successfully.
```

Figura 161: Creación de superusuario

Después, hay que ejecutar el siguiente comando para recuperar todos los ficheros estáticos que utiliza la aplicación de administración de la API:

```
python manage.py collectstatic
```

Por último, es necesario configurar el servidor web que se encargará de servir la aplicación. Como se ha comentado, hay que tener instalado Apache2 en el sistema, además de ciertos paquetes que se necesitan para soportar Python.

```
sudo apt-get install apache2-dev -y
sudo apt-get install apache2-mpm-worker -y
sudo apt-get install libapache2-mod-wsgi-py3
```

Después de instalar el servidor web, hay que editar el archivo de configuración para servir la API en Internet:

```
nano /etc/apache2/sites-available/000-default.conf
<VirtualHost *:80>
    ServerName docenciafdi.me
    ServerAlias www.docenciafdi.me
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
    Alias /static /home/TFG_docencia/static
    <Directory /home/user/TFG_docencia/static>
        Require all granted
    </Directory>
    <Directory /home/user/TFG_docencia/TFG_docencia>
```

```

        <Files wsgi.py>
            Require all granted
        </Files>
    </Directory>

WSGIDaemonProcess TFG_docencia python-
path=/home/user/TFG_docencia python-
home=/home/user/DocenciaEnv

WSGIProcessGroup TFG_docencia

WSGIScriptAlias /
/home/user/TFG_docencia/TFG_docencia/wsgi.py

WSGIPassAuthorization On

</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet

```

Finalmente, se procede a reiniciar el servidor web para que los cambios surtan efecto:

```
sudo service apache2 restart
```

Aunque no es obligatorio, se puede mantener una lista de hosts para los cuales permitimos que realicen peticiones a la API. Para ello se modifica la variable `ALLOWED_HOSTS=[]` añadiendo los hosts que se quieran añadir a la lista blanca. Actualmente se encuentra configurado para que cualquier host pueda realizar peticiones.

```
nano /home/user/TFG_docencia/TFG_docencia/settings.py
```